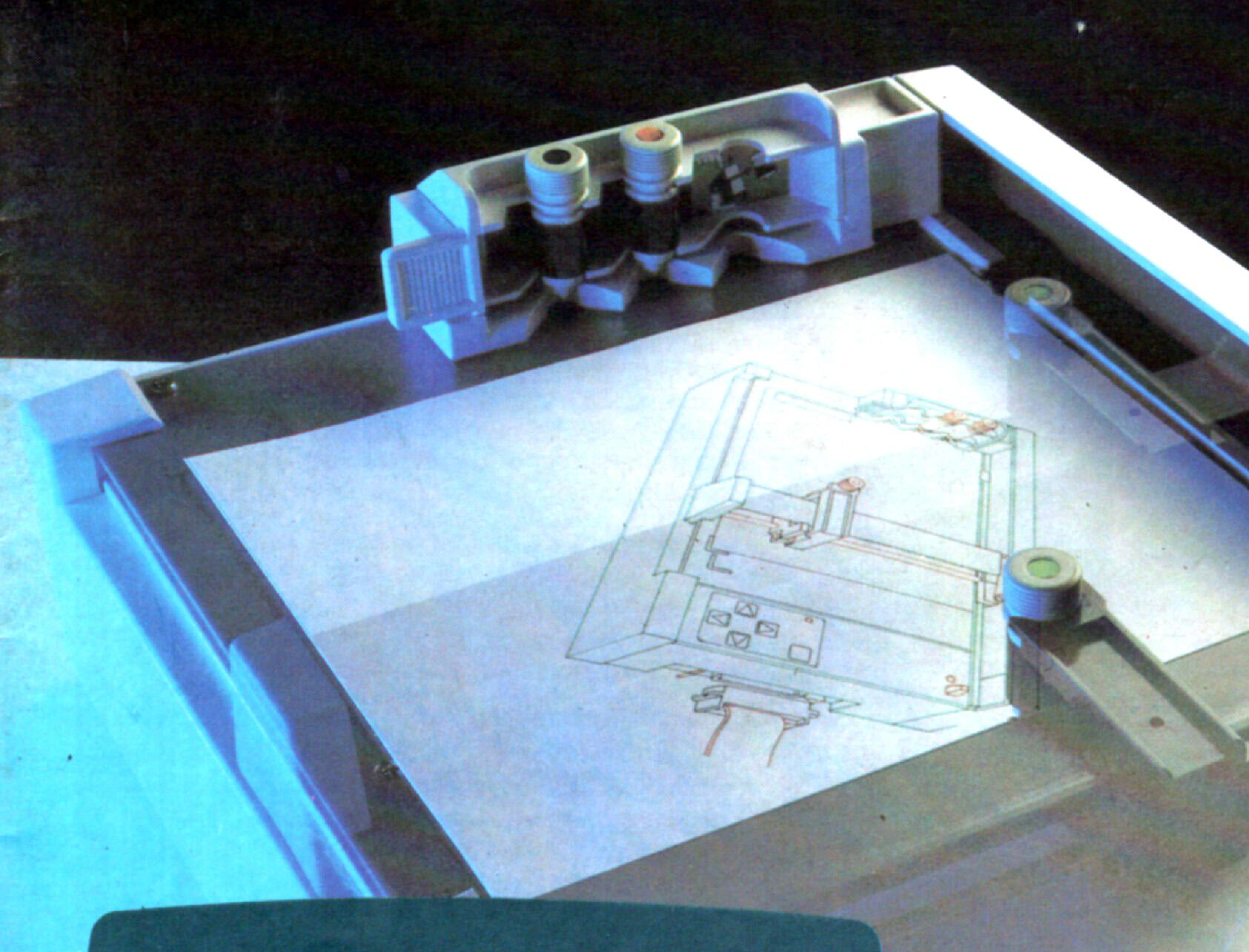
150 ptas.



CURSO PRACTICO DEL ORDENADOR PERSONAL, EL MICRO Y EL MINIORDENADOR



Editorial Delta, S.A.

MI COMPUTER CURSO PRACTICO
DEL ORDENADOR PERSONAL, EL MICRO Y EL MINIORDENADOR

Publicado por Editorial Delta, S.A., Barcelona, y comercializado en exclusiva por Distribuidora Olimpia, S.A., Barcelona

Volumen I - Fascículo 10

Director: José Mas Godayol
Director editorial: Gerardo Romero
Jefe de redacción: Pablo Parra
Coordinación editorial: Jaime Mardones
Asesor técnico: Roberto Quiroga

Redactores y colaboradores: G. Jefferson, R. Ford, S. Tarditti, A. Cuevas
Para la edición inglesa: R. Pawson (editor), D. Tebbutt (consultant editor), C. Cooper (executive editor), D. Whelan (art editor), Bunch Partworks Ltd. (proyecto y realización)

Realización gráfica: Luis F. Balaguer

Redacción y administración:
Paseo de Gracia, 88, 5.° - Barcelona-8
Tels. (93) 215 10 32 / (93) 215 10 50 - Télex 97848 EDLTE

MI COMPUTER, *Curso práctico del ordenador personal, el micro y el miniordenador,* se publica en forma de 96 fascículos de aparición semanal, encuadernables en ocho volúmenes. Cada fascículo consta de 20 páginas interiores y sus correspondientes cubiertas. Con el fascículo que completa cada uno de los volúmenes, se ponen a la venta las tapas para su encuadernación.
El editor se reserva el derecho de modificar el precio de venta del fascículo en el transcurso de la obra, si las circunstancias del mercado así lo exigieran.

ISBN: 84-85822-83-8 (fascículo) 84-85822-84-6 (tomo 1)
84-85822-82-X (obra completa)
Depósito Legal: B. 52-84

Fotocomposición: Tecfa, S.A., Pedro IV, 160, Barcelona-5
Impresión: Cayfosa, Santa Perpètua de Mogoda (Barcelona) 218403
Impreso en España - Printed in Spain - Marzo 1984

Editorial Delta, S.A., garantiza la publicación de todos los fascículos que componen esta obra.

Distribuye para España: Marco Ibérica, Distribución de Ediciones, S.A., Carretera de Irún, km 13,350. Variante de Fuencarral, Madrid-34.
Distribuye para Argentina: Viscontea Distribuidora, S.C.A., La Rioja 1134/56, Buenos Aires.
Distribuye para Colombia: Distribuidoras Unidas, Ltda., Transversal 93, n.° 52-03, Bogotá D.E.
Distribuye para México: Distribuidora Intermex, S.A., Lucio Blanco, n.° 435, Col. San Juan Tlihuaca, Azcapotzalco, 02400, México D.F.
Distribuye para Venezuela: Distribuidora Continental, S.A., Ferrenquín a Cruz de Candelaria, 178, Caracas, y todas sus sucursales en el interior del país.

Pida a su proveedor habitual que le reserve un ejemplar de MI COMPUTER. Comprando su fascículo todas las semanas y en el mismo quiosco o librería, Vd. conseguirá un servicio más rápido, pues nos permite realizar la distribución a los puntos de venta con la mayor precisión.

Servicio de suscripciones y atrasados (sólo para España)

Las condiciones de suscripción a la obra completa (96 fascículos más las tapas, guardas y transferibles para la confección de los 8 volúmenes) son las siguientes:

a) Un pago único anticipado de 16 690 ptas. o bien 8 pagos trimestrales anticipados y consecutivos de 2 087 ptas. (sin gastos de envío).

b) Los pagos pueden hacerse efectivos mediante ingreso en la cuenta 3371872 de la Caja Postal de Ahorros y remitiendo a continuación el resguardo o su fotocopia a Distribuidora Olimpia (Paseo de Gracia, 88, 5.°, Barcelona-8), o también con talón bancario remitido a la misma dirección.

c) Se realizará un envío cada 12 semanas, compuesto de 12 fascículos y las tapas para encuadernarlos.

Los fascículos atrasados pueden adquirirse en el quiosco o librería habitual. También pueden recibirse por correo, con incremento del coste de envío, haciendo llegar su importe a Distribuidora Olimpia, en la forma establecida en el apartado b).

Para cualquier aclaración, telefonear al (93) 215 75 21.

No se efectúan envíos contra reembolso.



Dibujos animados

Tome los gráficos de su ordenador, multiplique su calidad por mil, y obtendrá un sistema de animación por ordenador

Todo el proceso de realización de filmes, ya sea para el cine o la televisión, se basa en la incapacidad del cerebro de "congelar" una imagen. Una rápida sucesión de imágenes produce la impresión de movimiento.

Los primeros intentos de crear la ilusión de una imagen en movimiento se lograron mediante un cilindro ranurado, en cuyo interior se había pegado una tira de dibujos, que se hacía girar rápidamente. Al mirar a través de las ranuras, se ve una tosca representación de una imagen (o "fotograma") tras otra. El *zoótropo*, nombre que recibió este dispositivo, es un precursor de la ciencia de la fotografía, aunque, naturalmente, al poco tiempo los dibujos del interior del tambor fueran sustituidos por fotografías. El siguiente paso, el soporte fílmico, requería emulsiones fotográficas de acción relativamente rápida, capaces de grabar una imagen en menos de un dieciseisavo de segundo, puesto que los primeros filmes se proyectaban a 16 imágenes por segundo.

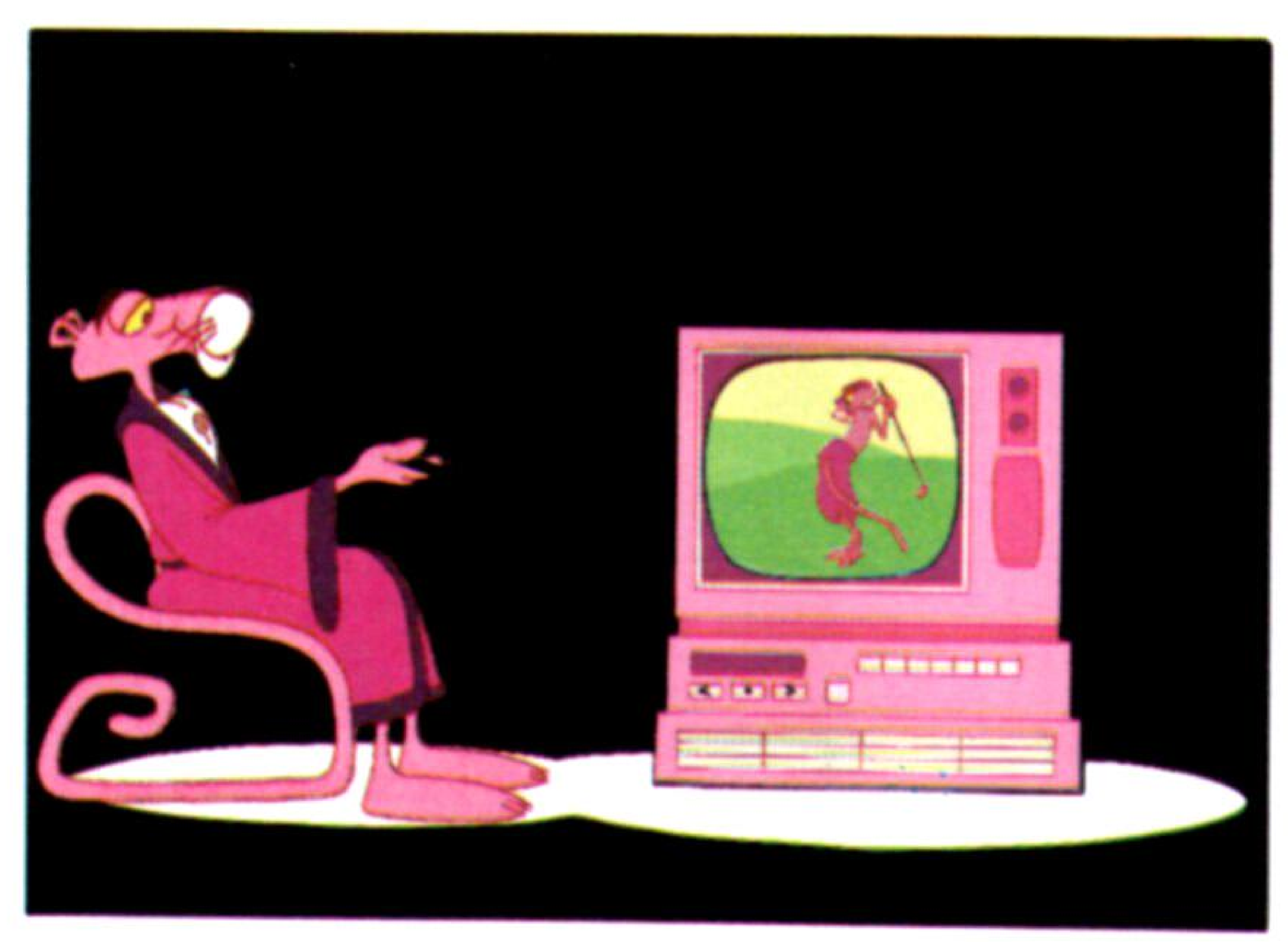
Técnicas de simulación del movimiento

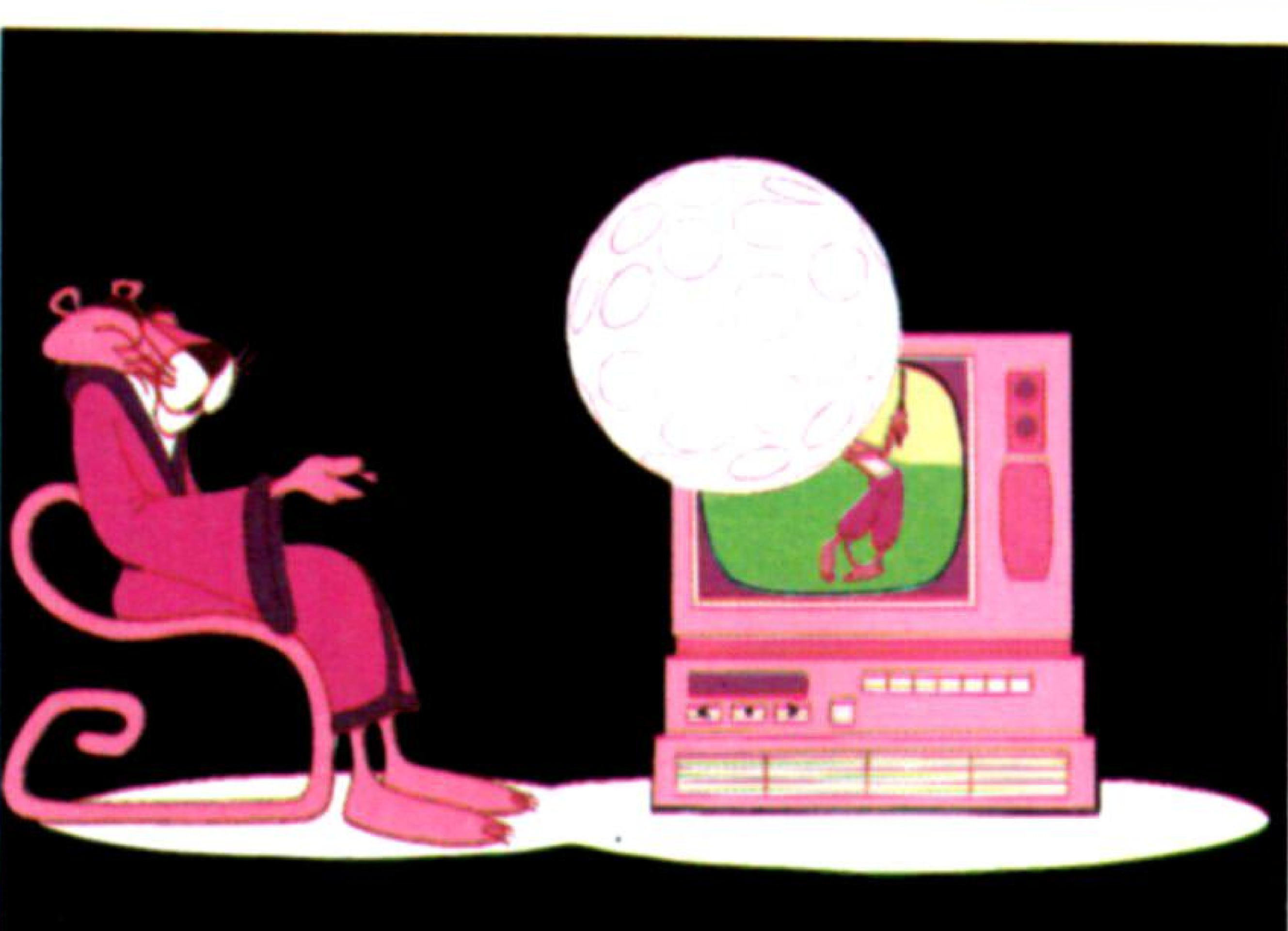
Extrañamente, algún tiempo antes, la industria cinematográfica concibió la idea de dibujar a mano cada fotograma, fotografiar los dibujos y proyectar el resultado para producir dibujos animados. Teniendo en cuenta que cada segundo de visión requiere la creación de 24 dibujos diferentes (la velocidad de proyección de una película moderna), está claro que la producción de un filme de tan sólo cinco minutos necesita una cantidad de trabajo enorme, 7 200 fotogramas en este caso. No es sorprendente, por tanto, que en el diseño de las ilustraciones se deban cumplir ciertas reglas: el requisito más importante es que se repitan con exactitud. ¡Sería inadmisible, por ejemplo, que en una película Bugs Bunny cambiara de apariencia a cada segundo!

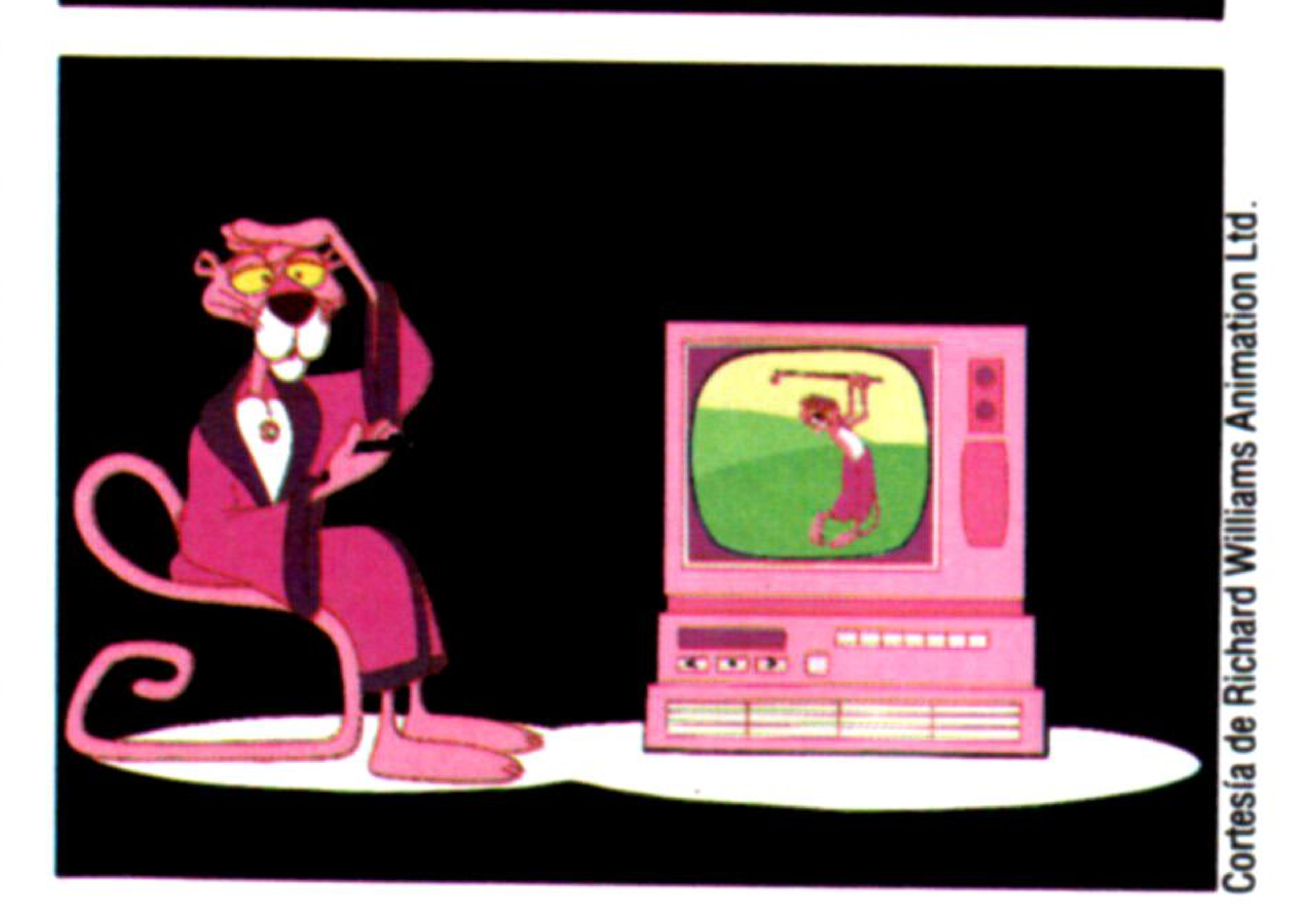
Tareas repetitivas y precisas similares a ésta son realizadas fácilmente por máquinas. Cuando el ordenador asume el control del trabajo de animación —ajustando la velocidad de movimiento, cambiando la perspectiva y geometría, aumentando la luz o la sombra, produciendo cambios en el volumen y el ritmo—, el artista puede concentrarse en la calidad de la imagen. De esta forma, la animación pasa a ser una verdadera arte gráfica, en la cual el artista emplea su tiempo en la creación de la imagen, que el ordenador hará que se mueva.

En su forma más simple, este proceso utiliza gráficos *sprite* (véase p. 152) para crear el "reparto"; los personajes son trasladados a la pantalla y puestos en movimiento, produciendo el tipo de animación empleada en los juegos de video sencillos. Para crear tanto la ilusión de cambio como la de movimiento (por ejemplo, alguien andando), es necesario sustituir repetidamente un sprite por otro. Como se vio, la creación de sprites es una tarea relativamente lenta si consideramos que la calidad gráfica del resultado y la imagen dependen sólo de una representación bidimensional muy sencilla.

La siguiente etapa de la animación requiere que el dibujante-programador construya un algoritmo que introduzca una sensación de profundidad en la imagen según las normas de la perspectiva. Así los objetos pueden definirse en la pantalla según sus coordenadas X, Y y Z. En este punto, es extremadamente práctico







Cortesía de Richard Williams Animation Ltd.

Fotograma a fotograma
Los dibujos animados convencionales, como estos fotogramas de la Pantera Rosa, exigen que el artista dibuje cada figura por separado, aunque no es necesario volver a dibujar los rasgos comunes, a menos que cambien su aspecto o posición. Se utiliza un soporte transparente, de forma que la imagen completa se obtiene a partir de una serie de superposiciones. El artista concentrará su atención en los fotogramas clave de la secuencia, dejando el acabado de las secciones menos importantes a cargo del equipo de ayudantes. Los dibujos acabados se fotografían con una truca, en el orden en que serán vistos

el que el programa no reproduzca líneas "ocultas" e introduzca opacidad en lo que ha sido, hasta ahora, un modelo o representación esquemática.

Otro refinamiento deseable es el "suavizar" las curvas. Una línea curva se determina con sólo tres puntos: los dos extremos y el punto más alejado de la línea recta que une ambos. Desde luego, una curva compleja (una S, por ejemplo) necesita ser descompuesta en sus componentes simples para así ser tratada adecuadamente. En este sentido es esencial disponer de algún sistema sencillo que indique al ordenador que la línea en cuestión es una curva que requiere ser suavizada y que no se trata sólo de una línea con un ángulo recto.

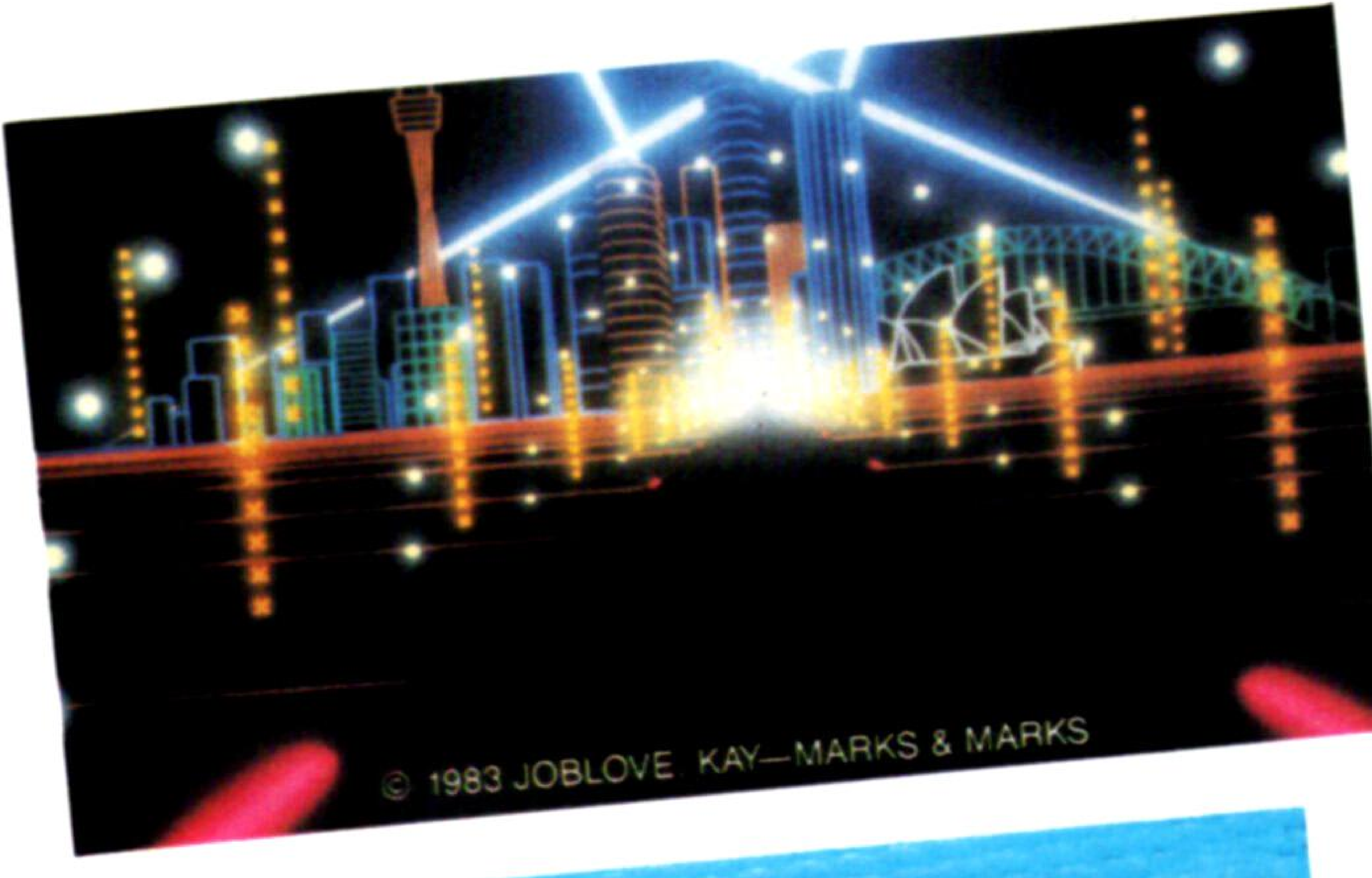
Otra característica es la capacidad de introducir luces y sombras en el dibujo. Ante todo es necesario especificar la posición de la fuente luminosa. La parte del objeto dibujado que queda frente al foco luminoso será más brillante y se irá oscureciendo progresivamente para ayudar a definir la forma del objeto. Un software sofisticado permitirá emplear más de una fuente luminosa y tratar la reflexión de la luz de un objeto a otro.

El uso del color tiene directa relación con el sombreado. Incluso el más simple de los ordenadores personales dispone en la actualidad de ocho o incluso dieciséis colores, pero los ordenadores gráficos para uso profesional proporcionan, como mínimo, 4 096. Algunos de ellos están limitados simplemente por el número de dígitos binarios del lenguaje del ordenador. Si éste es de 24 bits, por ejemplo, el ordenador tiene unas 16,7 millones de opciones de color. Los mandos de sombreado y coloreado están combinados en uno solo.

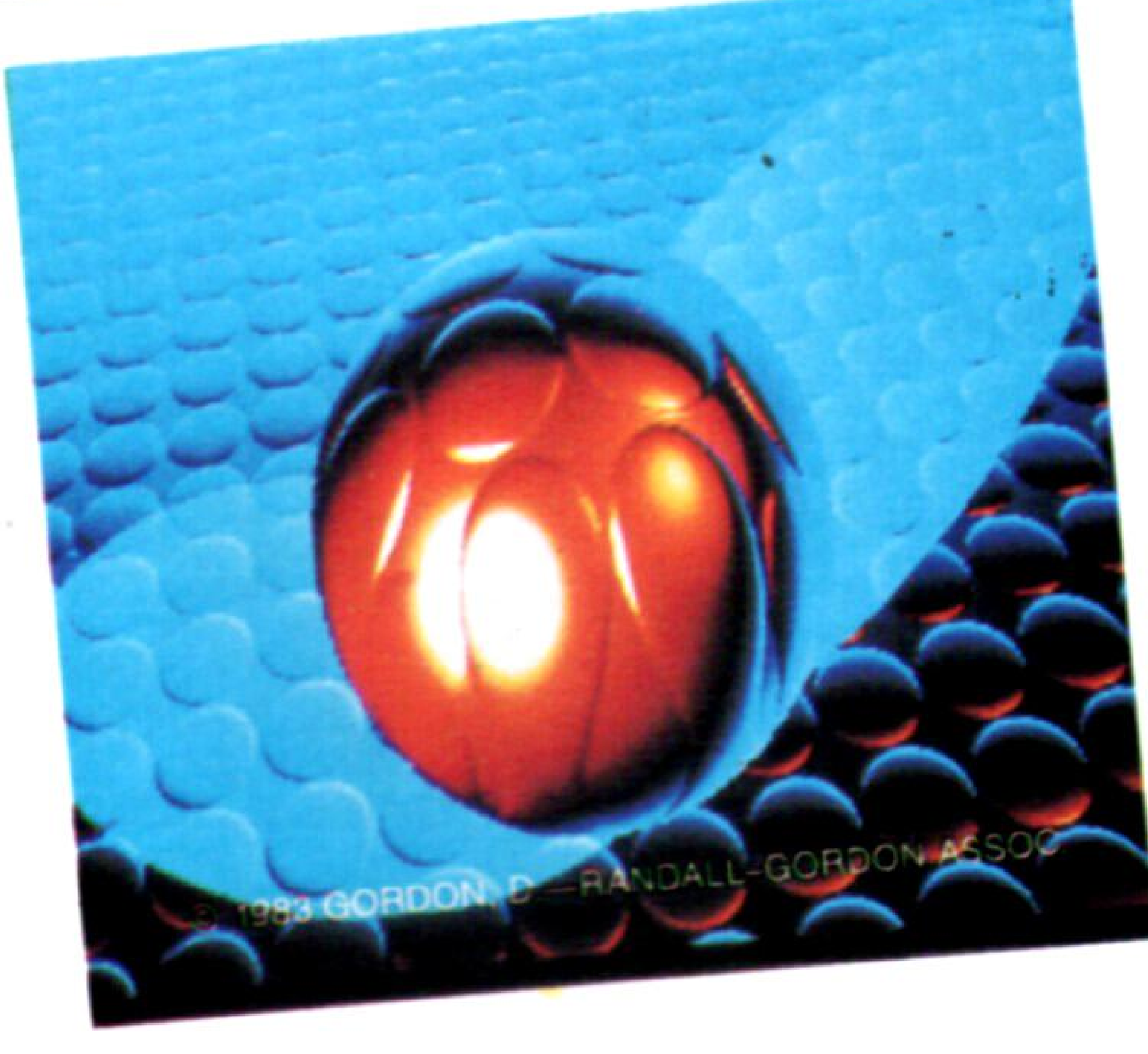
Veamos ahora el problema de simulación de movimiento. Resulta relativamente fácil dividir el movimiento de un objeto en sus componentes, si se le imagina como un problema de geometría de trazos continuos, incluso cuando el objeto representado es tan complejo como una mano humana. El factor determinante es el tamaño y la potencia del ordenador que se está utilizando. Hay que tener en cuenta, sin embargo, que para producir una imagen de alta calidad se necesitará un monitor capaz de dar una resolución de 1 000 × 1 200 pixels. Cada uno de estos pixels requerirá, como mínimo, un byte de ocho bits para tratar la información que define su color y brillo. Esto significa más de un megabyte por pantalla. En consecuencia, la generación de imágenes en movimiento de alta calidad no es posible en los ordenadores personales. La verdad es que los profesionales de este campo utilizan algunos de los ordenadores más grandes y potentes del mundo, y sus honorarios reflejan este hecho, pues un segundo de película acabada puede costar más de 200 000 pesetas.

Si se toma como punto de partida un objeto sencillo, semejante a un cubo, es relativamente fácil entender cómo se puede lograr que se mueva por la pantalla. Un cubo puede definirse sólo por las coordenadas de sus ocho vértices, pero exactamente el mismo principio se aplica a objetos más complejos. La única diferencia está en la cantidad de memoria necesaria para almacenar todas las coordenadas y la potencia de procesamiento que se precisa para poder manejar esa información con la suficiente velocidad para generar movimiento a "tiempo real". En esta aplicación, al igual que en las demás, existe la inevitable interdependencia entre calidad, cantidad de espacio disponible y resolución. Cuanto más pequeña sea la unidad de dibujo que se define, mayor será la necesidad de almacenamiento. Pero, para conseguir una alta calidad de reproducción, es esencial trabajar con el mayor grado de detalle posible.

Un televisor doméstico —con unos 625 × 1 000 pixels— produce imágenes con un "grano" más grueso





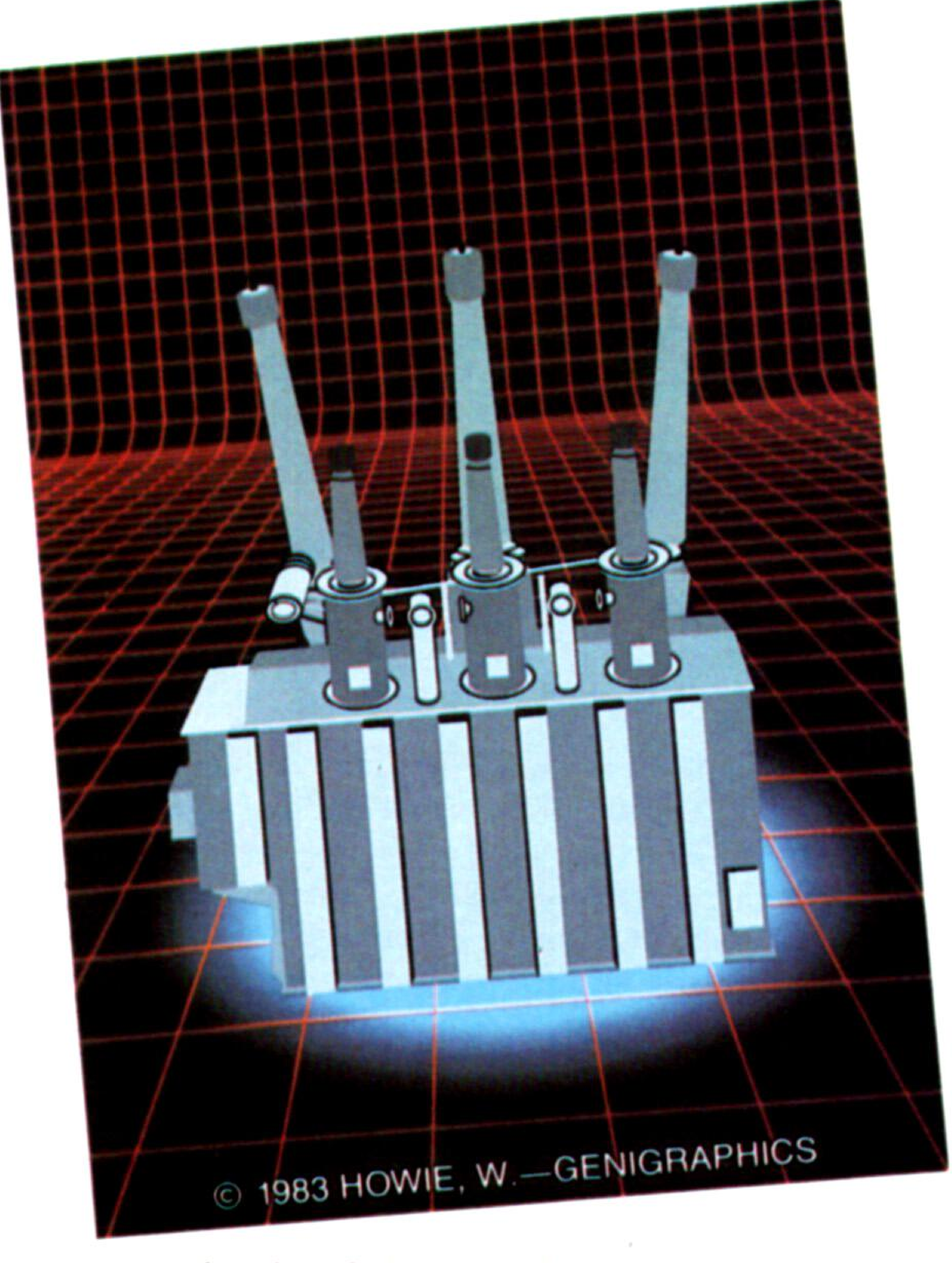




Caja de colores mágica
Hasta hace poco tiempo, el método de realización de vistas fijas y secuencias de dibujos animados para el cine y la televisión era muy parecido al que emplean las revistas: el dibujo se hacía sobre papel o una película transparente y luego era fotografiado. El sistema Quantel's Paint Box, sin embargo, elimina por completo el uso del papel componiendo el trabajo artístico digitalmente en el interior del ordenador y grabándolo directamente en la cinta de video

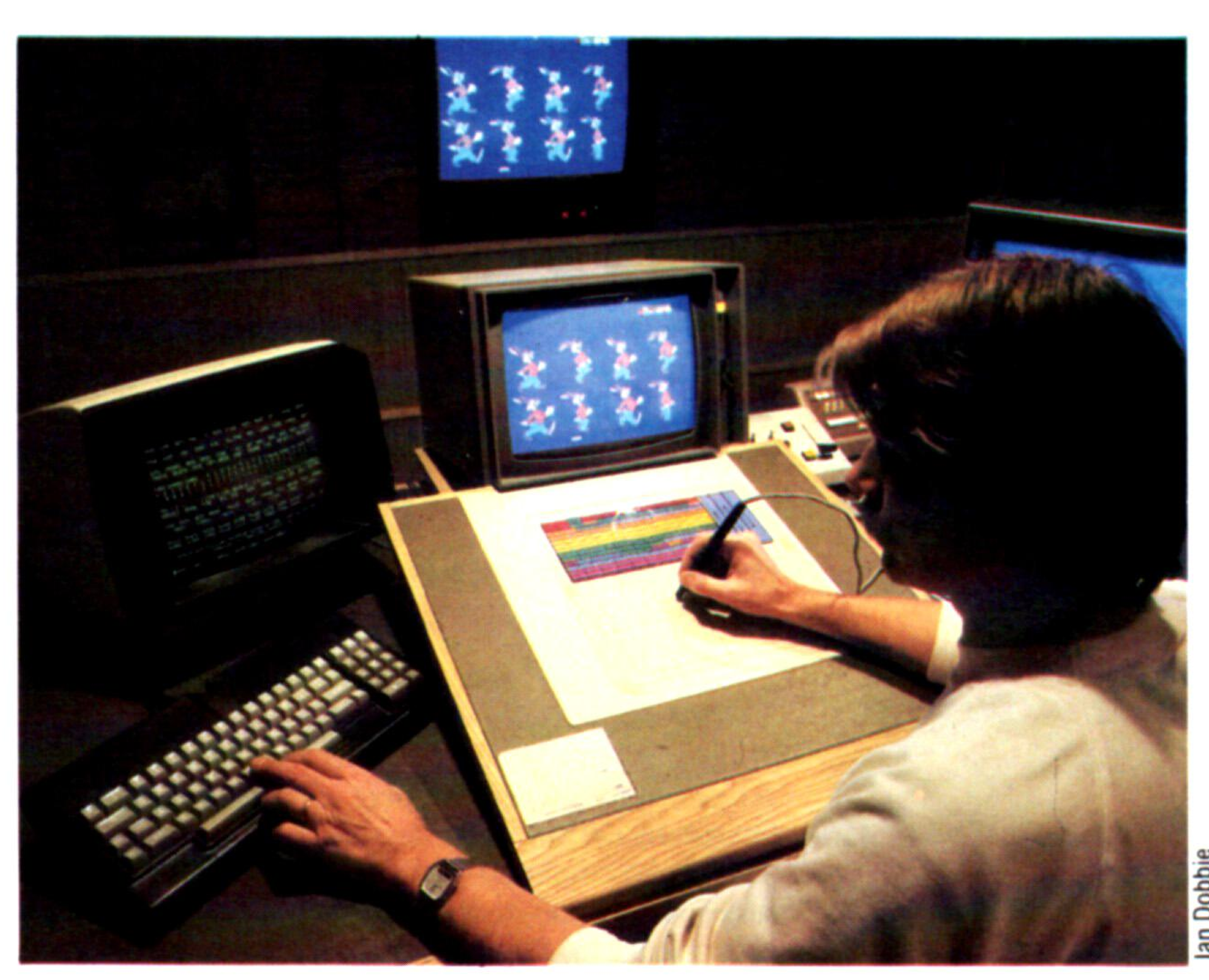
que el monitor de alta resolución del que se ha hablado anteriormente. Por tanto se puede estar seguro de que cualquier trabajo con esta o mayor definición será tan realista como una película corriente para televisión. Incluso, con técnicas adecuadas, se puede crear una impresión fiel de realidad mediante dibujos animados.

Para lograr una imagen con este grado de detalle se necesita un software sofisticado así como un hardware adaptado especialmente o fabricado para este fin. El método más popular usa un dispositivo denominado *digitalizador de bits*, el cual es bastante parecido a un gran tablero de dibujo que contiene un cuadriculado alámbrico. Esta malla se emplea para percibir la posición de un cursor que se desplaza a través de ella. Con un digitalizador de bits es posible realizar diversos cometidos: desde reproducir una obra de arte hasta dibujar a mano alzada o utilizar instrumentos de dibujo



convencionales, de la misma forma como si se estuviera dibujando sobre papel. La imagen es digitalizada (son determinadas sus coordenadas X-Y), escrita en la memoria y visualizada por el ordenador. El tipo del trazo que aparece en la pantalla puede definirse tal como se puede elegir entre usar un lápiz, una pluma o un pincel. Asimismo, puede definirse el color haciendo aparecer la paleta, que consiste en una gama de colores localizada en la parte inferior de la pantalla que se asemeja a la paleta de un pintor. Si el color que se desea no es estándar, puede mezclarse exactamente igual que si se usara pintura. El cursor también puede emplearse como goma de borrar, y los "dibujos" pueden superponerse.

Por tanto, habiendo creado una sola imagen, ¿cómo puede lograrse que se mueva? Simplemente mecanizando el proceso convencional, mediante un sistema de ordenador para clasificar el material, colorear siluetas e, incluso, para mostrar el resultado aproximado de una secuencia. Sin duda alguna, este método acelerará el trabajo, pero las técnicas de programación inteligente hacen posible lograr mucho más. Al igual que


Ian Dobbie

las curvas pueden ser suavizadas, también es posible realizar los bloques completos de acción, especificando el primer y el último fotograma de una secuencia. Este proceso, en un estudio de animación convencional, es llevado a cabo por un ayudante. Por supuesto, la mayor parte del trabajo de animación es realizado por ayudantes, y es a éstos a quienes sustituye el ordenador.

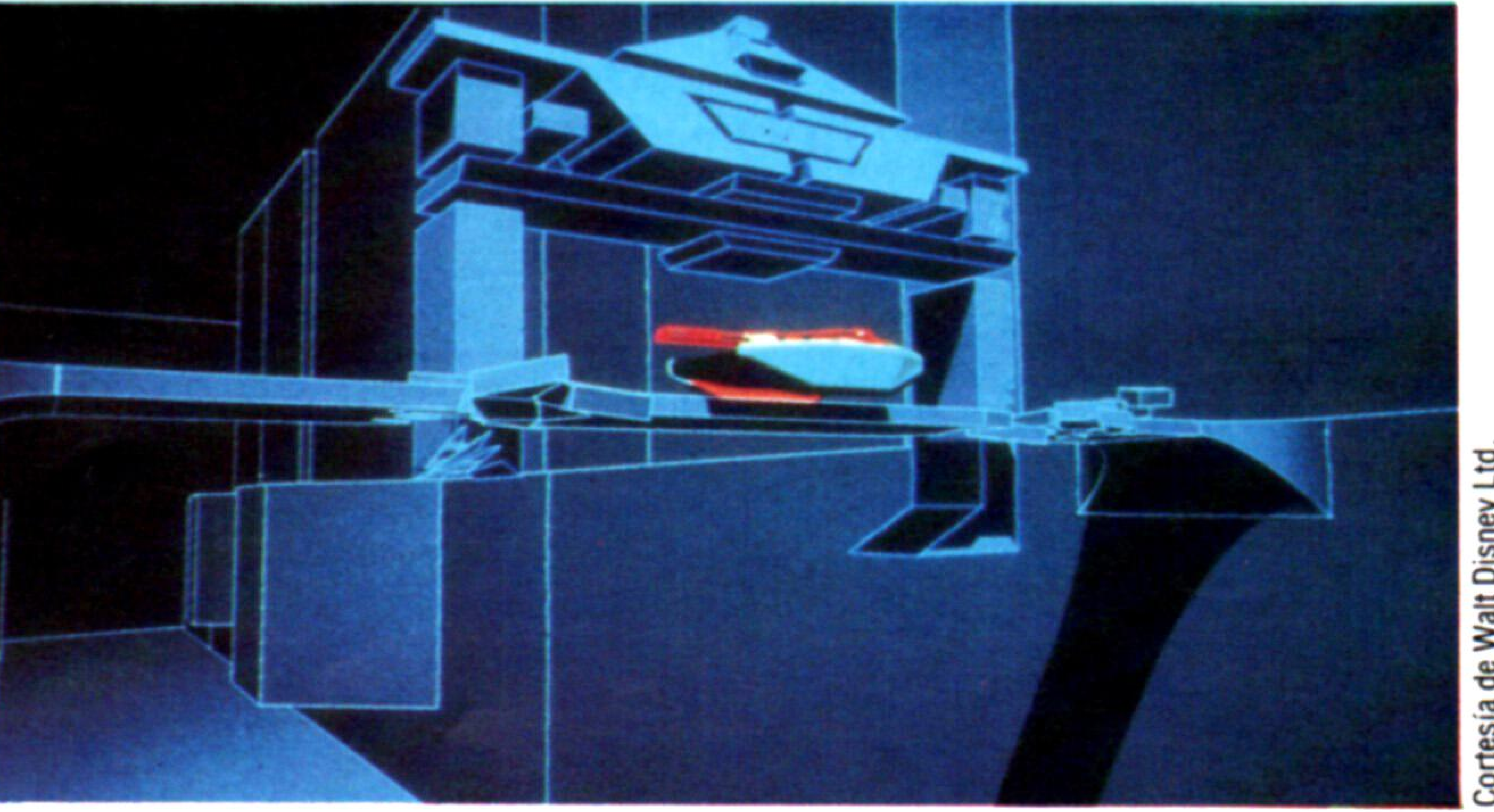
Anteriormente se puso de relieve que la introducción de los ordenadores en el proceso de animación deja libre al artista para que concentre su atención en la calidad de la imagen. La mayor parte del esfuerzo de un realizador de dibujos animados se destina a la labor de crear la ilusión de movimiento, pero esta tarea se puede definir con absoluta precisión en un ordenador.

Una vez establecidas las reglas, simplemente obedeciéndolas se producirá el resultado deseado, que constituirá, una vez más, una segura muestra de que el trabajo que se ha llevado a cabo es adecuado para las técnicas informáticas.

Nuestra pretensión con este artículo sobre la animación ha sido únicamente exponer otra de las innumerables aplicaciones del ordenador. No descartamos, sin embargo, abordar nuevamente el tema con mayor profundidad.

Ver es creer
Gracias al poder de procesamiento ultrarrápido y a la enorme capacidad de almacenamiento de los ordenadores modernos, es posible crear, en el cine o en la televisión, una imagen que no se puede distinguir de una fotografía. Luego, empleando técnicas de programación desarrolladas para la resolución de problemas estadísticos y numéricos, es posible manipular de tal forma estas imágenes creadas, que el espectador cree que son reales

Película pionera
TRON, de la Walt Disney Productions, fue el primer largometraje que empleó técnicas de generación de imágenes con ayuda de ordenador. Combinación de fotogramas rodados convencionalmente y de otros realizados en el interior de un ordenador gigante, esta película mezcló imágenes animadas por ordenador y fotografía de efectos especiales para crear un mundo de fantasía


Cortesía de Walt Disney Ltd.

La traducción alternativa

Los ordenadores "piensan" según el código máquina; los programadores prefieren escribir en un lenguaje de alto nivel como el Basic. Compiladores e intérpretes ofrecen métodos de traducción que difieren de unos a otros

Al principio los ordenadores no tenían teclado. Las instrucciones del programa tenían que introducirse paso a paso situando cada uno de los ocho interruptores en las posiciones *up* o *down* para representar una sola operación. Estas dos combinaciones constituían el código de la máquina.

Era lógico, entonces, reemplazar los interruptores por un teclado tipo máquina de escribir y sustituir el código por palabras de uso corriente. El resultado fue el lenguaje de "alto nivel", el BASIC, por ejemplo, que ha desplazado a los códigos de lenguaje máquina de bajo nivel.

Como procesadores, sin embargo, los ordenadores no cambiaron, sino que continuaron funcionando con los interruptores originales (y aún lo siguen haciendo). Por ello los programadores tuvieron que desarrollar software escritos en la notación original de bajo nivel para traducir esos programas de alto nivel a una forma que los procesadores pudieran manipular. Estos programas de bajo nivel recibieron la denominación de *compiladores* o *intérpretes*, de acuerdo a su método de traducción.

En informática (como en cualquier otra disciplina), todo aumento de potencia o velocidad debe pagarse —en dinero, tiempo o libertad de acción—. Así sucede con los intérpretes y compiladores. Conjuntamente suministran todas las traducciones de programa que se necesitan. Los intérpretes son más adecuados para unas áreas y los compiladores para otras, pero cada uno paga por sus ventajas con desventajas compensadoras.

Los intérpretes, que normalmente se montan en los ordenadores personales, son la forma más barata de traducir programas de alto nivel a un lenguaje que el ordenador pueda entender. No emplean mucha memoria y dejan más espacio para los programas.

Los micros de precio más asequible utilizan casi siempre un intérprete BASIC: se escribe un programa en BASIC, se teclea RUN, y el programa funciona o se detiene e indica un error del sistema, algo parecido a:

```
ERROR DE SINTAXIS EN LINEA 123
```

Entonces de digita LIST, se halla el error, se corrige, se pulsa RUN y el programa funcionará o se volverá a parar, y así sucesivamente. Hay que tener en cuenta que algunos de los intérpretes BASIC más sofisticados verifican los errores de sintaxis al introducir cada línea.

Lo anterior puede haberse realizado cientos de veces sin haber pensado ni un solo momento en el intérprete. Su virtud principal es precisamente el ser un dispositivo invisible que permite trabajar en el programa sin tener que preocuparse de en qué lugar de la memoria se encuentra localizado o cómo ejecutarlo: el programa está en la punta de los dedos, y puede procesarse, listarse o editarse de inmediato.

El intérprete suele ser fácil de usar, siempre que no sea muy sofisticado: cada vez que se digita RUN, el intérprete tiene que encontrar en la memoria el programa en BASIC, traducirlo y realizarlo línea a línea. Si el programa contiene este bucle:

```
400 LET N = 0
500 PRINT N
600 LET N = N + 1
700 IF N < 100 THEN GOTO 500
```

el intérprete debe traducir y realizar las líneas 500 a 700 cien veces, como si nunca las hubiera procesado.

Los compiladores son diferentes. Son caros, difíciles de escribir, y ocupan y usan mucha memoria. Por lo general forman parte de un software soportado en discos; por tanto se necesita un sistema caro.

Lo que ofrecen es flexibilidad, potencia y velocidad; frente a las cuatro líneas de BASIC anteriores, un compilador las traduciría de una vez, luego realizaría ese código cien veces.

Esto permite ahorrar bastante tiempo, pero tiene un precio. Supóngase que se posee un compilador BASIC y se quiere introducir y procesar un programa en este lenguaje.

Primero se carga y se pasa un programa de creación de archivos (llamado *editor*), que permite introducir el programa mediante el teclado y guardarlo en disco como *archivo fuente*.

A los archivos se les debe dar un nombre para poder encontrarlos una vez se los ha creado (al igual que sucede con un archivo clásico); por lo tanto, el programa editor pide que se les dé una denominación. Estos nombres por lo general constan de dos partes: la primera es una etiqueta, cualquier nombre que se elija (por ejemplo, MYPROG), y la segunda normalmente es un código de tres letras que indica la naturaleza de los datos contenidos en el archivo; este código se denomina *extensión*. En BASIC, podría tener el código BAS. El archivo fuente está ahora en disco bajo el nombre de MYPROG. BAS. Entonces, al escribir el usuario:

```
COMPILE MYPROG. BAS
```

hará que el ordenador cargue y ejecute el compilador BASIC con un archivo denominado MYPROG. BAS.

Compilación de un programa
Escribir un programa compilado no es, ni con mucho, tan sencillo como usar el intérprete de un ordenador personal. Sin embargo, una vez el programa esté funcionando, será ejecutado con mucha mayor rapidez. Éste es el organigrama principal:

Cargar el programa "editor"

Introducir el programa en BASIC

Guardar el programa como "archivo fuente" en cassette o disco

Cargar el programa "compilador" y dar nombre al archivo fuente

El compilador produce un "archivo objeto": versión del archivo fuente en código de máquina

Si hubiese errores en el programa, éstos serán comunicados, y debe reeditarse el archivo fuente

Cuando ya no haya errores, el archivo objeto puede ser cargado directamente en la máquina y procesado

El compilador tarda unos segundos, según sea su longitud, en traducir el programa a un *archivo objeto*, que guarda un disco bajo el nombre de MYPROG.OBJ. La extensión OBJ indica que éste es un archivo objeto, es decir, una traducción a código de máquina de un archivo fuente.

Mientras el compilador traduce el archivo, verifica asimismo sus errores de sintaxis. Si encuentra alguno, enviará un mensaje similar a éste:

```
100 REED X:IF X=3(N + 2) LET P=Q
      1              2      3
FATAL ERROR:-
1) //REED// UNRECOGNISED COMMAND
2) //(//ILLEGAL OPERATOR HERE
3)??'THEN' OR 'GOTO' EXPECTED HERE
```

Este mensaje se produce para cada línea que contiene un error. En otras palabras, la información sobre el error es mucho más extensa que la que proporciona un intérprete BASIC.

Llegados a este punto, hay que volver a cargar y pasar el editor, extraer del disco el archivo fuente, realizar los cambios e intentar compilar otra vez. Si no hay más errores, se puede escribir:

```
RUN MYPROG
```

y el programa funcionará, tal como se espera, o no. En esta etapa ya no hay ningún error de sintaxis, porque el programador ya los ha corregido; no obstante, puede darse el caso de que éste quisiera introducir cambios en el programa, para realizar lo cual este último se carga y se pasa el editor, se cambia el archivo fuente, se recopila... y así sucesivamente.

Las virtudes de un compilador no son evidentes en la fase de desarrollo del programa, aunque las informaciones sobre errores son valiosas. Éstos empiezan a adquirir su real importancia cuando ya se tiene un programa de trabajo y se digita la orden RUN.

Los programas compilados son rápidos. Suelen serlo de 5 a 50 veces más que los programas por intérprete, según sea la eficacia del compilador, aunque la velocidad de ejecución del programa compilado tiene su contrapartida en la fase del desarrollo del programa.

Si se comparan compiladores e intérpretes con respecto a las secuencias descritas más arriba, los compiladores quedan en desventaja, puesto que éstos, por lo general, están escritos para máquinas más potentes y menos especializadas, en las cuales se pueden escribir y procesar los programas en otros muchos lenguajes.

El COBOL (lenguaje de alto nivel, para escribir programas de proceso de datos comerciales para control de cuentas, nóminas e inventarios), por ejemplo, fue inventado pensando en utilizar un compilador, mientras que el BASIC requiere realmente un intérprete.

Una vez se ha desarrollado y compilado un programa, el archivo fuente ya no se necesita más que como referencia. En consecuencia, el programa fuente puede ser totalmente comentado y escrito con el propósito de que sea legible, mientras que el archivo objeto puede ser mucho más pequeño y ocupar menos espacio en el disco y la memoria.

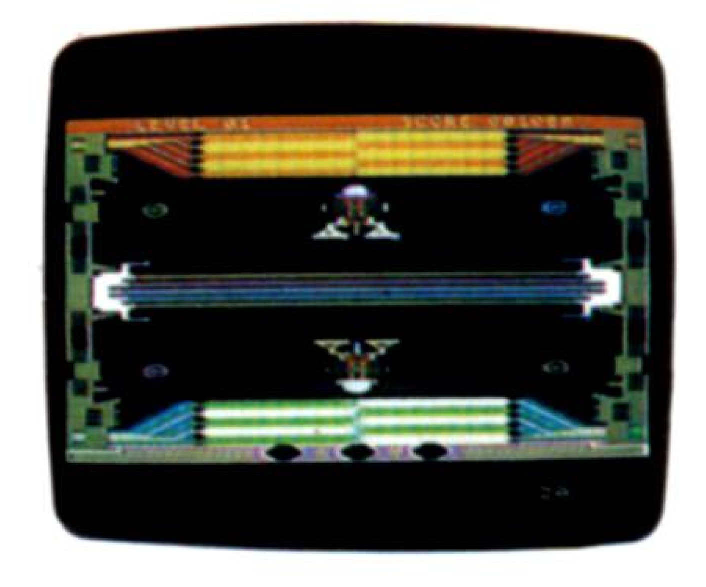
El hecho de que los archivos objeto creados por un compilador se compongan de código de máquina ilegible, puede ser, sorprendentemente, una ventaja. Si se comercializa software, no se venderá el archivo fuente, sino sólo el archivo objeto, lo cual hará que sea mucho más difícil hacerlo objeto de piratería, copiarlo o alterarlo.



Lento
En un juego de aventuras, la velocidad no es primordial, y la mayor parte del programa consiste en la manipulación de un texto. Éste está escrito en BASIC, y es interpretado mientras se procesa



Más rápido
Muchos programas de gestión (en particular, hojas electrónicas) son difíciles de escribir en código máquina, debido a que contienen muchas operaciones matemáticas. Sin embargo, un lenguaje de intérprete sería demasiado lento. Por ello, frecuentemente, están escritos en BASIC y luego son compilados



El más rápido
Para los juegos recreativos de acción rápida, que requieren gráficos, incluso un programa compilado sería moroso. Estos paquetes deben escribirse directamente en código de máquina: una tarea lenta y engorrosa

Reproducir el habla

Formaba parte de la ciencia-ficción. Pero, en la actualidad, con un sintetizador de voz, un ordenador puede verdaderamente hablar con usted

Mientras la ciencia de la identificación de voz todavía tiene que desarrollarse totalmente, la generación del habla electrónica ya es una realidad. Hasta hace poco tiempo, sin embargo, los factores esenciales para producir una pronunciación semejante a la humana eran la potencia del ordenador y la capacidad de memoria. Hoy en día, con la ayuda de complementos adecuados, casi todos los ordenadores personales y juguetes electrónicos son capaces de contestar.

Hay dos formas de generar electrónicamente sonidos semejantes a la voz humana. El primero, hasta hace poco tiempo el más común, es el de la síntesis mediante normas. Analizando las frecuencias contenidas en el habla, es posible idear un sistema de reglas que permita crear, a partir de sus componentes, cualquier sonido dado. Por ejemplo, la palabra *mar* podría definirse por el número de milisegundos de la mezcla de frecuencias que crean el sonido m, seguido inmediatamente por las frecuencias de a y r.

Estos bloques de formación individuales se llaman *fonemas* y mediante combinaciones de ellos se puede construir cualquier palabra. Al generar la voz a través de este procedimiento, las características propias de la persona que habla tienden a perderse, pero las palabras pueden reconocerse y entenderse claramente. Debido a que las reglas para generar los fonemas están formadas en el interior del equipo, el usuario puede introducir una lista de los fonemas en el sistema. Éstos son reproducidos por medio de un pequeño altavoz. Con un poco de práctica es posible generar frases completas inmediatamente utilizando secuencias de fonemas, las cuales, por lo general, son susceptibles de ser almacenadas en órdenes de BASIC.

El segundo método de síntesis de voz se basa en la capacidad que poseen el oído y el cerebro para completar los intervalos. Por ejemplo, la gama de frecuencias que puede transmitirse por una línea telefónica sólo proporciona una quinta parte de la calidad que podría esperarse de un equipo corriente de alta fidelidad; no obstante, la voz que se oye a través del auricular es perfectamente inteligible. Ello se debe a que nuestro cerebro completa los intervalos.

El método de síntesis denominado *voz digitalizada* se sirve del mismo fenómeno. Con la reducción de coste de la memoria de ordenador, ahora se puede convertir el habla en información digital mediante un convertidor analógico-digital. Luego los datos resultantes son comprimidos cientos de veces y almacenados en una ROM, creando de ese modo los intervalos que después puede compensar el oído.

Para lograr que cualquiera de las palabras almacenadas sea pronunciada, sólo hay que proporcionar al ordenador la dirección de esa palabra en la ROM, y la información digital es recuperada y convertida otra vez en sonido. Debido a que se han almacenado las palabras originales de la persona que hablaba, se mantienen las características personales de la voz.

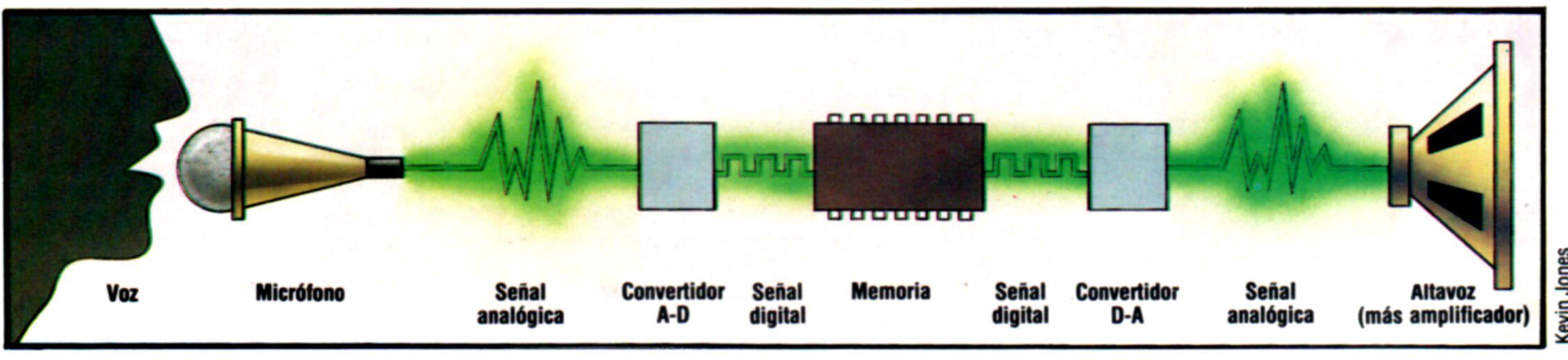
Algunos ordenadores poseen hardware y software en discos que permiten al usuario digitalizar su propia voz empleando un micrófono. Los datos resultantes se almacenan en disco (un segundo de conversación ocupa alrededor de 1 Kbyte) y pueden extraerse, cuando sea necesario, de un programa de aplicaciones en forma de mensajes verbales o avisos.

Los usos de los sintetizadores de voz son tantos y tan variados que es prácticamente imposible proporcionar una lista completa de ellos. Para empezar, pueden reemplazar los avisos grabados en cinta magnetofónica de las estaciones de ferrocarril, aeropuertos y otras terminales.

Las unidades de síntesis de voz han sido incorporadas en algunos modelos de automóvil como un componente estándar del tablero de instrumentos. Y esto no es sólo una estratagema de ventas, pues el sintonizador avisa realmente al conductor y éste puede tomar las medidas necesarias sin tener que apartar la vista de la carretera.

En el mercado de los ordenadores personales y de los juegos electrónicos, los sintetizadores de voz se emplean para mejorar sus características: para anunciar la puntuación, para avisar de ataques del enemigo, permitiendo que el jugador se concentre en las tácticas en vez de tener que consultar mensajes impresos en la parte inferior de la pantalla.

Por último, existen dispositivos didácticos tales como *Speak and spell* (Hable y deletree), que recita una palabra que debe ser deletreada correctamente, y diccionarios de idiomas que pronuncian las palabras al aparecer en la pantalla. Más avanzada la obra, trataremos de nuevo el tema con mayor detalle.



El flujo del sonido
La voz puede ser digitalizada y almacenada en memoria, tanto en RAM como en ROM. La señal eléctrica de salida de un micrófono pasa a través de un convertidor analógico-digital. La salida de este chip es en forma digital de unos y ceros. La voz puede recrearse mediante un convertidor digital-analógico, un amplificador y un altavoz

Imagen tridimensional

El cine y la televisión han popularizado en estos últimos tiempos las imágenes en tres dimensiones. Un ordenador puede crear visualizaciones muy eficaces en este sistema

La mayor parte de la gente cree que para que la televisión tridimensional se convierta en realidad antes se habrá de producir algún descubrimiento trascendental. Y esto no es exacto. Con un ordenador es posible producir imágenes tridimensionales de calidad bastante aceptable en un televisor en color corriente.

Eso sí, estas imágenes no son totalmente en color; sólo se visualizan los bordes y las esquinas de un objeto, pero se podrá percibir una verdadera profundidad.

Debe dejarse claro que el término *tridimensional* se aplica con frecuencia para describir programas disponibles a nivel comercial. Pero en la mayoría de los casos este adjetivo no es empleado apropiadamente: el producto que se ofrece es en realidad un juego "plano en pantalla" común y corriente, al que se le ha dado cierta perspectiva o agregado algún sombreado.

Sin embargo, la genuina visualización tridimensional incorpora una tercera dirección de movimiento, hacia afuera y hacia adentro. Esto se consigue utilizando un par de gafas bicolor, una de cuyas lentes es roja y la otra de color azul verdoso. El sistema en tres dimensiones se ha empleado en ciertos libros de ilustraciones y en algunas películas espectaculares de reciente producción como *Jaws-3 (El gran tiburón)*.

La forma de saber si se está mirando una imagen tridimensional falsa o auténtica es a través del color. Las verdaderas visualizaciones tridimensionales no tienen mayor sentido si se las mira sin las gafas, en cuyo caso consistirán mayormente en una serie de líneas trazadas en dos colores diferentes, rojo y azul verdoso. En conjunto, estas líneas constituirán una confusa mezcla de intersecciones. Una imagen que contenga cualquier color menos estos dos, y en particular una imagen que se pueda apreciar con facilidad, probablemente no sea una auténtica imagen tridimensional. Incluso aunque en su etiqueta diga que lo es, el juego carecerá de una verdadera profundidad. El movimiento sólo será posible en dos planos: de arriba abajo y de derecha a izquierda.

Es bastante sencillo producir una imagen que convenza a sus ojos y a su cerebro de que está mirando una escena auténticamente tridimensional, si bien para ello han de entrar en juego las matemáticas.

El primer requisito consiste en proporcionar una imagen ligeramente diferente para cada ojo. Dado que los ojos verán la escena desde posiciones alejadas, las dos imágenes tienen que dibujarse con unos ángulos y una perspectiva ligeramente distintos. Estas representaciones no deben interferirse, aunque ambas estén en la misma pantalla. Esto se logra dibujando una imagen en rojo y la otra en azul verdoso.

Si la imagen se contempla a través de un par de gafas bicolor en las que el filtro de color azul verdoso esté colocado frente al ojo izquierdo, la imagen roja (dirigida al ojo derecho) será casi invisible para el ojo izquierdo. El mismo principio pero a la inversa se aplica para la imagen azul verdosa (dirigida al ojo izquierdo). Mediante este proceso podemos aislar las dos imágenes. El resultado será una imagen tridimensional bastante convincente. Este sistema ofrece el inconveniente de que, aunque hemos utilizado el color para aislar entre sí las dos representaciones, la escena es efectivamente monocromática. Al agregarle rojo al azul verdoso se produce una imagen más o menos blanca, que es lo que su cerebro percibe.



Tony Sleep

Juegos en relieve
El británico Mike Singleton se ha pasado tanto tiempo usando gafas con colores rojo y azul verdoso mientras desarrollaba sus juegos tridimensionales, que quizá ahora para su cerebro sea normal procesar esas imágenes coloreadas. En cualquier caso, Singleton verdaderamente ha dado una nueva "dimensión" a los juegos para ordenadores personales al aplicar a los gráficos una técnica del cómic que ya tiene 30 años de existencia

Otro pequeño problema se hace evidente si consideramos lo que sucedería cuando un objeto se haya de mover "hacia dentro" o "hacia fuera" de la pantalla. Un objeto situado a una distancia de tres metros no aparecerá con un tamaño dos veces más grande que otro que esté a una distancia de seis metros. Los tamaños se deben calcular mediante trigonometría, lo cual se puede realizar empleando la función tangente (TAN en la mayoría de las versiones de BASIC).

El cálculo de los valores y su conversión en coordenadas para el trazado en pantalla supone diversos cálculos con números decimales. Esto es bastante lento y por ello hay un límite respecto al nivel de complejidad que puede alcanzar una visualización o a la rapidez con que ésta se puede mover. La visualización de muchos objetos y la realización de los consiguientes cálculos pueden retardar mucho el proceso, hasta el punto de que el movimiento se perciba espasmódico y lento. Además, para aislar las dos imágenes una debe estar separada de la otra por al menos un punto, de modo que ambas puedan ser de distinto color.

Un factor importante en la producción de imágenes tridimensionales es la calidad de la visualización en pantalla. Se logran mejores resultados con un monitor que con un televisor, pues en el primero los colores son más claros y con márgenes más nítidos.

Peek y Poke

Estas dos órdenes se utilizan cuando se quiere programar algo a lo que no puede hacer frente el lenguaje Basic, pero cada máquina las emplea de modo diferente

POKE
Es necesario utilizar con cuidado la orden POKE, puesto que cambia el contenido de las localizaciones de memoria, y ello podría afectar al funcionamiento del ordenador, y causar la pérdida de un programa. Se puede probar con las siguientes órdenes "seguras" POKE.
En el Atari 400 u 800, si mediante POKE se introduce 1 en la posición 751, el cursor de la pantalla desaparecerá. Pruebe POKE 751,1.
En el Commodore 64, pruebe POKE 1024,1. 1024 es la dirección de la primera localización de pantalla.
En Spectrum, probar:

```
100 FOR N = 0 TO 6
    STEP 2
110 POKE USR "A" + N,
    BIN01010101
120 POKE USR
    "A" + N + 1,
    BIN10101010
130 NEXT N
140 PRINT "AAAAAAAA"
```

Las A de la línea 140 deben introducirse a través del módulo de gráficos. Al procesar el programa se producirá una línea de símbolos de tableros de ajedrez en miniatura. Sin embargo, también puede producir formas de interferencia en el televisor

PEEK y POKE son dos órdenes del lenguaje BASIC que se emplean en programas avanzados, que necesitan manipular bits y bytes individuales en la memoria. La orden PEEK se usa para examinar el contenido de una dirección específica (localización) de la memoria, y la orden POKE se utiliza para almacenar un número (del 0 al 255) en un punto específico de la memoria.

Mediante ellas el programador puede acceder al funcionamiento interno del ordenador de un modo que no sería posible con otro sistema. Normalmente el BASIC propio del ordenador es el que se ocupa de saber los puntos donde se encuentran almacenados los datos y las variables que definen los caracteres a visualizar en la pantalla. Aunque, por regla general, el operador no debe preocuparse en saber en qué parte de la memoria se hallan localizados estos datos, a veces sí puede ser necesario. La orden PEEK permite hacerlo.

Fácilmente puede escribirse un programa corto para examinar cualquier punto de la memoria:

```
10 REM BUSCANDO LOCALIZACIONES DE MEMORIA
20 PRINT "INTRODUCIR LOCALIZACION DE MEMORIA
   EN DECIMAL"
30 INPUT M
40 P = PEEK(M)
50 PRINT "CONTENIDO DE LOCALIZACION";M;
   "ESTA";P
60 GOTO 20
70 END
```

Esto imprimirá el contenido de la dirección específica, expresada en forma de número decimal. (En realidad, por supuesto, el ordenador lo almacena en binario.) Si se quiere ver a qué es equivalente el contenido en términos de caracteres "imprimibles", el lenguaje BASIC incluye una función que convierte los números decimales en sus caracteres equivalentes. Ésta es la función CHR$, y cambiando la línea 50 se imprimirán los equivalentes de los caracteres:

```
50 PRINT "CONTENIDO DE LOCALIZACION";M;
   "ESTA";CHR$(P)
```

Para examinar la totalidad de la memoria, puede añadirse un bucle FOR...NEXT, suprimiendo la línea 30, cambiando la línea 20 por FOR X = 0 TO 65535 y sustituyendo la línea 60 por NEXT X.

Si se quiere tener tiempo suficiente para ver cuándo se imprime cada carácter, quizá sea necesario añadir un bucle de retraso tras la orden PRINT y antes de NEXT X. Téngase en cuenta también que el límite superior del bucle FOR...NEXT presupone que se tiene una memoria de 64 Kbytes. Este número puede cambiarse para memorias más pequeñas: 16 Kbytes requiere, en notación decimal, 16383; 32 Kbytes necesita 32767, y 48 Kbytes precisa 49151. El desarrollo completo de este programa es:

```
10 REM PEEKING AND PRINTING TODAS LAS
   LOCALIZACIONES
20 FOR X = 0 TO 65535
30 LET Y = PEEK(X)
40 PRINT "LOCALIZACION";X;"=";Y;"=";
50 PRINT CHR$(Y)
60 FOR D = 1 TO 200
70 NEXT D
80 NEXT X
90 END
```

Aunque la función CHR$ convierte los números decimales en sus equivalentes, los caracteres imprimibles se representan con los números 32 a 127. La mayor parte de los ordenadores utilizan los números entre 128 y 255 (el número mayor representable en un solo byte) para caracteres gráficos especiales.

Muchos de los números entre el 0 y 31 desempeñan funciones especiales de control de pantalla. Al pasar el programa, cuando éstos son encontrados en la memoria, serán convertidos, por la CHR$, en efectos de pantalla especiales, por ejemplo, harán que la pantalla se vuelva negra o que el cursor se desplace a la esquina superior izquierda.

La orden POKE es, en esencia, opuesta a PEEK. Permite "escribir" un byte o dato (cualquier número del 0 al 255) en un punto de la memoria. Debe tenerse cuidado al emplear POKE; si se utiliza para introducir un número y se hace en una sección errónea de la memoria, podría destruir o alterar parte del programa esencial. La única forma de remediar este hecho sería arrancar de nuevo el ordenador (desconectándolo y volviendo a conectarlo otra vez, a menos que tuviera un mando de borrado), y esto comporta el riesgo de destruir alguno de los programas. Por consiguiente, antes de utilizar la orden POKE, se debe consultar el manual para hallar en el esquema de la memoria un área designada como *área para el usuario*.

En muchos ordenadores personales, la memoria video (la empleada para almacenar los caracteres que se visualizarán en la pantalla) es accesible al usuario. Normalmente, el ordenador recibe la información sobre la forma de los caracteres a visualizar desde un circuito ROM especial, llamado generador de caracteres, el cual almacena las estructuras de puntos de cada carácter. Pero, por lo general, también es posible emplear una RAM. Si los códigos de las estructuras se almacenan en RAM, se podrán introducir, mediante la orden POKE, nuevas estructuras, expresadas como números decimales, en las direcciones apropiadas de la RAM, y usarlas para definir caracteres visualizables totalmente nuevos.

TI99/4A

El ordenador personal de la Texas Instruments es como un Mercedes entre Volkswagen. La calidad de construcción es elevada, pero los complementos son caros

Desde el punto de vista del diseño y construcción, el TI99/4A de la Texas Instruments es uno de los ordenadores personales más profesionales. Es una máquina que tiene tras de sí la experiencia de una firma con un prestigio reconocido hace mucho tiempo en el mundo de la informática.

Utiliza un microprocesador de 16 bits, el TMS9900, diseñado y fabricado por la misma Texas Instruments, compañía que también fabrica semiconductores, calculadoras, microprocesadores y miniordenadores. El microprocesador TMS9900 fue uno de los primeros chips de 16 bits, pero no logró obtener una amplia popularidad.

El TI99/4A tiene un teclado con 48 piezas, que, comparado con el nivel general de los ordenadores personales, resulta muy cómodo. A su derecha existe un espacio destinado a los cartuchos de software, denominado por la firma fabricante como software de "estado sólido". Una conexión similar, situada en el borde derecho del aparato, permite adaptar una extensión de hardware. Los módulos de ampliación, que son grandes cajas de plástico, contienen controladores de unidades de disco, ampliación de memoria y una interface serial (RS232). Se conecta por medio de una caja de expansión, unidad considerada esencial si se desea ampliar la máquina.

La pantalla posee 16 colores con gráficos de alta resolución; el generador de sonido es capaz de producir tres notas o "voces" independientes al mismo tiempo. Sin embargo, la falta de buena documentación hace que sea difícil aprender a escribir los programas en código de máquina, el uso de gráficos y la utilización de las características de sonido.

Texas Instruments vende directamente sus equipos, así como a través de distribuidores, y prácticamente sólo existen los periféricos que suministra la misma firma.

El ordenador TI99/4A está diseñado para poder ser utilizado por principiantes en informática. Su lenguaje es el BASIC, y el más popular como complemento es el LOGO. En Estados Unidos es muy empleado en los centros escolares, compitiendo con el Apple II para el primer puesto en el concepto de ventas entre los ordenadores didácticos.

Cuando se conecta, en la pantalla aparece un menú que ofrece al usuario un número variado de eleccio-



Chris Stevens

Teclado del TI99/4A
Es de un nivel más elevado que el de la mayoría de los ordenadores personales, aunque varios usuarios han comentado que el "bote" de cada tecla es demasiado duro. El número de teclas es también bastante limitado, presumiblemente para dejar sitio a la carga de cartuchos, en el lado derecho. La mayoría de las teclas, por tanto, tiene dos funciones: pulsando "CTRL" y "E" se logra que el cursor se desplace hacia arriba. La tecla "FCTN" convierte la hilera superior de teclas en definibles por el usuario, y es posible insertar una cinta plástica sobre esta hilera, en la cual puede escribirse la designación para cada pieza

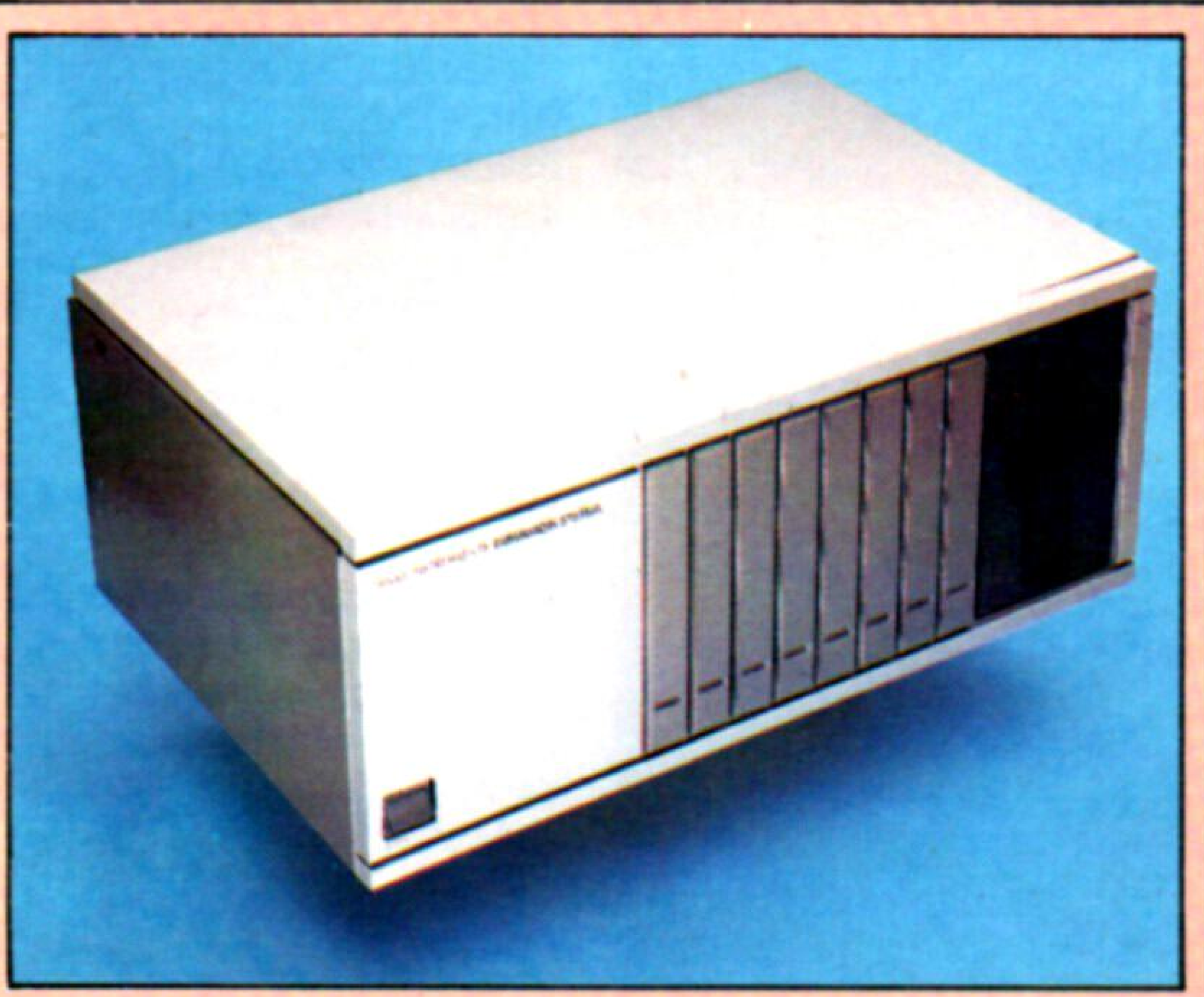
nes. Si se introduce un cartucho de software en el ordenador, el operador tiene la posibilidad de pasar el nuevo software o el BASIC. Las cualidades del BASIC incorporado en el ordenador son limitadas, pero cabe la posibilidad de adaptar un cartucho BASIC "ampliado", con el cual se enriquecen sus potencialidades (mejorando incluso las que ofrece un Microsoft estándar) y se proporcionan órdenes de impresión normalizadas (véase p. 53), gráficos "sprite" y la posibilidad de manejar un sintetizador de voz. Para el funcionamiento de este último es necesario el BASIC ampliado o un cartucho editor de voz.

El TI99/4A posee numerosas ampliaciones de software y hardware. Existe todo tipo de periféricos y puede adquirirse una variada gama de lenguajes de programación.

El sistema de este ordenador, en su conjunto, es de fácil manejo para los principiantes, y su sólida construcción lo hace adecuado para ser utilizado incluso por niños de corta edad.



La palanca de mando
Las palancas de mando de la Texas Instruments (designadas *controladores por cable*) están formadas por un par de unidades, conectadas a un enchufe para su adaptación al ordenador. En cada dispositivo hay cuatro interruptores, que se asemejan a las conexiones de la parte inferior de algunos teclados



Caja de ampliación periférica
Esta caja dispone de una fuente de energía, conexiones y espacio para los módulos para ampliación de memoria, interfaces de discos e impresora. Estos módulos están formados por cajas de plástico que contienen circuitos con un conector terminal en la base, un testigo óptico de conexión en la parte frontal y diversos cables procedentes de la cara posterior. En la caja hay capacidad para ocho cargas. La primera de la izquierda contiene el módulo que conecta la caja de ampliación al ordenador, y la de la derecha está destinada al módulo electrónico de la unidad de disco. Quedan seis cargas para la memoria y la ampliación de la conexión serial. Sólo se puede añadir un módulo extra de memoria de 32 Kbytes, con una RAM máxima de 52 Kbytes. El módulo interface serial permite conectar dispositivos en serie como impresoras y modems de formato RS232

Chris Stevens

Conexión video
Esta conexión suministra las señales básicas para la generación de los sistemas de televisión PAL (España y casi toda Europa occidental) y NTSC (EE. UU.)

RAM
La máquina sale de fábrica con una RAM de 16 Kbytes, que puede ampliarse externamente

Conexiones palanca de mando
Esta conexión única de patillas múltiples permite empalmar las dos palancas de mando, fabricadas también por Texas

Componentes discretos
Otra característica de ordenadores semejantes a éste, que fueron diseñados hace varios años, es su gran número de componentes discretos: transistores y resistencias. Actualmente, un chip puede sustituir a docenas de ellos

ROM
La ROM de la máquina puede ser complementada con la conexión de cartuchos. Por ejemplo, al ampliar el BASIC se incrementará el número de mandos disponible

Interruptor
Incorpora un indicador LED de conexión



Conexión cassette
El TI99/4A puede funcionar con dos grabadoras de cassette corrientes, y controlar el motor de una de ellas. Esto significa que puede ejecutar programas comerciales sin acabar, que requieren la copia de datos de una consola a otra

CPU
El TMS9900 es uno de los primeros procesadores, lo cual explica su gran tamaño. Todas las líneas de dirección y datos, así como las líneas de control, tienen conexiones independientes. Los procesadores más modernos poseen patillas para varias funciones y por ello un chip puede tener menor número de ellas. Es un microprocesador de 16 bits

Conexión periféricos
Es sólo un conector terminal PCB, al que se acoplan otras unidades. Texas lo denomina como interface CRU (*Communications Register Unit*: unidad de registro de comunicaciones). Antes de que se introdujera una unidad ampliadora de uso general, cada periférico se conectaba en serie al siguiente en una larga línea

Conexión paquete ROM
Los paquetes ROM, que Texas llama *módulos de mando de estado sólido*, se conectan en este punto. El mecanismo es mucho más sólido que en la mayor parte de las máquinas

Memoria "scratchpad"
Los chips marcados con 6810 constituyen una memoria *scratchpad* ("bloc de apuntes"), esencial para el funcionamiento del 9900. Este microprocesador es diferente de todos los demás al no tener localizaciones de memoria interna (registros) y por ello necesita memoria externa. Ésta no es accesible a los programas normales

TI99/4A

DIMENSIONES
380 × 260 × 70 mm

PESO
1,8 kg

VELOCIDAD DEL RELOJ
1 MHz

MEMORIA
ROM: 28 Kbytes; RAM usuario: 16 K; RAM gráficos: 8 K. Existen 256 bytes extra de RAM *scratchpad*, normalmente no accesibles al usuario. Se emplean para registros internos del 9900; la mayoría de CPUs lo llevan incorporado

VISUALIZACION EN VIDEO
Visualiza caracteres en 24 líneas de 32 columnas. Existen 16 colores, que pueden utilizarse como primeros planos o fondos. En la máquina básica no se dispone de gráficos de usuario, pero, con una secuencia de 16 caracteres, pueden definirse celdas individuales de 8 × 8 caracteres

INTERFACES
Cassette, palanca de mando, video (no TV), carga de cartuchos y conexión para el bus de ampliación

LENGUAJE SUMINISTRADO
BASIC

OTROS LENGUAJES DISPONIBLES
BASIC ampliado, TI LOGO, UCSD (University of California at San Diego), PASCAL, TI FORTH y Assembler

VIENE CON
Adaptador toma de potencia, adaptador TV, conexión cassette y manuales

TECLADO
Tipo máquina de escribir con 48 teclas móviles, incluyendo teclas de control y función. Según los cartuchos de software que se introducen, las teclas numéricas lo son también de función

DOCUMENTACION
Viene acompañado de un manual principal que describe cómo conectar el ordenador y usar los "módulos de mando sólido". Esta introducción es muy corta y tiene diagramas, pero no fotografías. Hay una lista detallada de las funciones disponibles en BASIC, una sección con ejemplos de algunos programas y un pequeño glosario al final

Piratas en acción

La piratería es una espina clavada en la industria del software, y los distribuidores adoptan severas medidas para combatirla



Copiador esclavo
Las cassettes de software, al igual que las de música, se duplican mediante un copiador de alta velocidad. Consiste en una consola en la cual se coloca la cassette original y en un número de unidades esclavas que efectúan las grabaciones simultáneamente. Copiar ambas caras de una cassette de programa es sólo cuestión de segundos. Los discos tienen que copiarse uno a uno utilizando unidades de disco corrientes

Ciento a uno
Del mismo modo que es ilegal realizar copias de cassettes de música propiedad de otras personas, efectuar copias de programas representa incurrir en piratería de software. Desgraciadamente para los distribuidores, ésta no sólo es difícil de evitar, sino también de detectar y procesar. Algunos fabricantes afirman que por cada ejemplar vendido legalmente se realizan cien copias ilegales

La piratería de software puede definirse, simplemente, como la copia de programas no autorizada. Al igual que en la industria del disco, con la cual tiene muchos problemas en común, la piratería se produce de diferentes formas y a distintos niveles. Al nivel más bajo, se comete piratería cada vez que el propietario de un ordenador personal copia un programa que le ha prestado un amigo. Incluso el hecho de que algunos programas (especialmente los escritos en código de máquina) no puedan conservarse con las órdenes normales de BASIC, no constituye un impedimento importante, puesto que siempre será posible conectar entre sí dos cassettes y copiar el programa, sin necesidad de emplear un ordenador.

Algunos distribuidores de juegos afirman que por cada ejemplar vendido, se realizan hasta cien copias ilegales. Aunque se pueda argüir que algunos fabricantes están en condiciones de afrontar estas pérdidas, debe recordarse que muchas personas obtienen su medio de vida gracias a los royalties de los programas, ¡y no viajan en Rolls Royce!

Se ha producido una amplia controversia sobre el hecho de que los distribuidores ofrezcan software con los sistemas de préstamo, alquiler o "pruebe antes de comprar", puesto que éstos facilitan la copia de programas. Algunos vendedores con menos escrúpulos van más lejos y regalan copias piratas de títulos populares a todo comprador de un ordenador, con el fin de incrementar su valor efectivo.

Existen casos incluso de distribuidores que reproducen programas a gran escala y los venden a otros —con menos riesgo del que parece si éstos son de otro

país—. Estos productos son equivalentes, por tanto, a las copias de "contrabando" de álbumes discográficos de gran popularidad.

A partir de este punto, la piratería se vuelve más sofisticada y más difícil de precisar. Por ejemplo, alguien toma un programa que ya existe, efectúa algunas modificaciones en él y lo pone a la venta como propio. La nueva versión puede ofrecer una mejora sustancial en rendimiento o en sus características, o simplemente puede presentar un cambio en los títulos de crédito que aparecen al pasar el programa por primera vez, o bien alterar la disposición de la información en la pantalla para que el paquete no sea reconocible inmediatamente. Este método se utiliza con mayor frecuencia en programas de gestión que en software de juegos.

Se puede discutir si el proceso de modificación de un programa puede considerarse como piratería de software, es decir, como si fuera una simple copia. Ésta es la causa de que mucha gente lleve a cabo esta labor impunemente. Los editores de programas están muy poco protegidos por la justicia y las leyes existentes sobre derechos de copyright no protegen a los programas de la piratería y de las modificaciones. Parece ser que los derechos de copyright sólo son aplicables a material impreso (con excepciones especiales para la música), y, en consecuencia, los programas de ordenador, que únicamente están almacenados en RAM o en cassette, no están amparados por estos derechos. Como en otros muchos temas legales, debe establecerse una jurisprudencia, y esto requiere, lógicamente, tiempo y dinero.

El punto más controvertido radica en el hecho, por ejemplo, de que una compañía tome una idea de un programa popular y reproduzca su propia versión. Nótese que no se está copiando ninguno de los códigos de programa, lo único que se hace es tomar notas precisas de cómo aparece el programa en la pantalla y de cómo reacciona a las órdenes que se le suministran, y luego empezar desde cero para conseguir el mismo efecto. El ejemplo más sobresaliente sobre esto ha sido el caso de PacMan, que empezó por ser un juego recreativo de máquinas tragaperras, fue puesto a la venta por la casa Atari para sus ordenadores personales y en cartuchos de video y posteriormente realizado en diferentes versiones por varios editores de software. Cada una de las versiones era ligeramente diferente, pero todas presentaban como atracción principal a la misma pequeña y familiar criatura que engulle ávidamente su camino alrededor del laberinto. Después de varios meses, Atari consiguió eliminar a la mayoría de estos competidores, ya fuera mediante la acción de la justicia o bien, en casos menos importantes, simplemente con la amenaza de llevar el caso a los tribunales.

Por lo general, los autores y distribuidores de software poseen medios, además de los legalmente establecidos, de proteger sus códigos de programa y royalties. Algunos distribuidores son partidarios del plausible punto de vista de que si venden sus productos a un precio suficientemente bajo, la gente tendrá menos interés en copiarlos. En programas más sofisticados, un manual bien editado o una presentación atractiva aportan un cierto grado de protección.

El "registro del usuario" es un método con el que se protegen softwares de gestión más caros: hasta que no se envía la tarjeta incluida en el manual del usuario, no se ofrece ninguna ayuda o asistencia técnica por vía telefónica.

Los llamados métodos "duros" de protección, normalmente, consisten en un dispositivo del tamaño de una caja de cerillas, denominado *dongle*, que debe ser adaptado a una de las conexiones de las interfaces del ordenador para que pueda procesarse el programa. Los circuitos del dongle incorporan un corto programa electrónico, normalmente un esquema de unos y ceros grabado en una de las ROM. Con cierta frecuencia el programa de aplicaciones se dirige al dongle; si no recibe la contestación del código correcto, rehúsa continuar. El código puede muy bien ser específico para cada dongle, lo cual significa que cada copia del programa sólo funciona con el dongle con el que se vende. La única forma de hacer copias ilegales es fabricar este dispositivo, o volver a escribir el código de programa para suprimir las secciones que hacen referencia a aquél. Esto no es imposible de realizar, pero está más allá de las posibilidades de la mayoría de los programadores personales.

Se han realizado muchas investigaciones con el fin de obtener una protección similar, sin tener que utilizar hardware adicional. La idea, acertadamente conocida como *filigrana*, consiste en tener un código magnético superpuesto en la cassette o el disco, "detrás" de la grabación del propio programa, y que no se transfiere al efectuar una copia. Por tanto el programa sólo funcionará con el disco o cassette originales.

La única protección "dura" económicamente viable que pueden utilizar los distribuidores de juegos por ordenador es el cartucho de ROM, el cual por lo general se vende a precios altos porque evita los largos periodos de tiempo de carga de cassettes. Sin embargo, ni los cartuchos son inexpugnables. En la actualidad existen dispositivos que pueden copiar un cartucho en una cassette, o en una especie de cartucho que el usuario puede programar o reprogramar.

La piratería del software es una batalla del tipo "policías y ladrones" en la cual los protagonistas tratan constantemente de superar en ingenio e inventiva a sus rivales en el mercado. No obstante, parece prácticamente imposible que se pueda eliminar; en el mejor de los casos puede conseguirse que sea lo suficientemente costosa como para convertirla en una actividad marginal.



Dongles
Existen unos pequeños dispositivos de hardware que se utilizan para proteger de la copia ilegal ciertos programas. Éstos no funcionarán a menos que se conecte el dongle adecuado en las interfaces del ordenador. Generalmente, sus circuitos electrónicos están recubiertos con resina sólida, y por tanto son muy difíciles de manipular

Ian McKinnell

Una nueva dimensión

Las matrices unidimensionales, como se ha visto, almacenan cierta cantidad de datos que tienen una característica común. Las matrices bidimensionales, por su parte, se utilizan para confeccionar tablas y cuadros

Hasta ahora se han considerado dos tipos de variables: simples y subíndice. Las primeras son semejantes a posiciones de memoria, en las cuales pueden almacenarse y manipularse números (o series de caracteres) refiriéndose a la variable "etiqueta". Las variables simples únicamente pueden almacenar un valor o serie y tienen nombres de variable "simple" —por ejemplo, N, B2, X, Y3—. Las variables subíndice, a veces llamadas matrices unidimensionales, pueden almacenar una lista entera de valores o series. El número de valores o series que tienen cabida se especifica al principio del programa mediante la orden DIM. Por ejemplo, DIM A(16) significa que la matriz denominada A puede contener 16 valores independientes. Debe tenerse en cuenta, sin embargo, que muchos lenguajes BASIC aceptan A(0) como primer elemento, y por consiguiente DIM A(16) define en realidad a 17 elementos. Para hacer referencia a estas "posiciones" se debe emplear el subíndice apropiado. PRINT A(1) imprimirá el primer elemento de la matriz: LET B = A(12) asigna el valor del decimosegundo elemento de la matriz a la variable B; LET A(3) = A(5) traspasa el valor del quinto elemento al tercero.

Algunas veces, sin embargo, es necesario poder manejar datos cuya mejor forma de representación es mediante tablas. Adviértase que tienen una gran semejanza a una hoja electrónica (véase p. 158). Los datos a representar pueden ser muy diversos, abarcando desde los resultados de los partidos de fútbol hasta la clasificación de ventas de unos grandes almacenes por artículos y departamentos. Como ejemplo de una tabla de datos típica, véase el siguiente desglose de los gastos de una familia durante el período de un año:

	Alquiler	Teléfono	Electricidad	Alimentación	Coche
ENE	20 000	3 800	3 500	52 700	8 300
FEB	20 000	3 500	2 900	51 000	6 800
MAR	20 000	3 400	3 100	49 900	7 000
ABR	20 000	3 300	3 000	50 000	9 100
MAYO	20 000	3 400	4 300	47 000	7 900
JUN	20 000	3 600	3 000	48 600	6 100
JUL	20 000	3 500	3 200	47 000	8 500
AGO	20 000	3 400	3 100	49 500	7 400
SEPT	20 000	3 300	3 000	46 300	6 900
OCT	20 000	3 700	3 800	52 800	7 000
NOV	20 000	3 900	3 800	53 000	7 500
DIC	20 000	4 100	4 700	55 900	8 600

El disponer la información de esta manera permite que sea tratada de forma relativamente simple. Es fácil, por ejemplo, hallar el total de los gastos del mes de marzo con sólo sumar todas las cantidades que figuran en la línea correspondiente a este mes. Igual de sencillo resulta hallar los costes anuales de teléfono y del automóvil sumando las columnas verticales. De forma semejante, no resulta difícil conocer los promedios mensuales o anuales de gastos por dichos conceptos. Esta tabla recibe el nombre de matriz bidimensional. Consta de 12 filas y 5 columnas.

En BASIC, las matrices bidimensionales semejantes a la anterior pueden ser representadas casi de la misma forma que las unidimensionales. La diferencia radica en que, para hacer referencia a su situación, ahora la variable requiere dos subíndices.

Si se estuviera escribiendo un programa en BASIC utilizando esta tabla de información, lo más sencillo hubiera sido tratar toda la tabla como una matriz bidimensional. Al igual que a una matriz ordinaria de una dimensión, se le atribuye un nombre, por ejemplo A. Y al igual que en aquéllas, será necesario DIMensionarlas. Como existen 12 filas y 5 columnas, se dimensionará así: DIM A(12,5). Es importante el orden en que se escriben los dos subíndices; la norma es que se coloquen las filas en primer lugar y las columnas en segundo. La tabla representada más arriba tiene 12 filas (una para cada mes) y cinco columnas (una para cada tipo de gasto), siendo por tanto una matriz de 12 por 5.

La orden DIM desempeña dos funciones esenciales. Deja aparte, en la memoria del ordenador, las posiciones necesarias para la matriz, y permite que cada una de estas posiciones sea determinada por el nombre de la variable seguido, entre paréntesis, por las posiciones de las filas y columnas. La orden DIM X(3,5), por ejemplo, crearía una variable X capaz de representar una matriz con tres filas y cinco columnas.

Véase la tabla y supóngase que la información ha sido introducida como si fueran los elementos de una matriz bidimensional denominada A. Hállense los valores de A(1,1), A(1,5), A(2,1), A(3,3) y A(12,3).

Es posible introducir una tabla de información como si se tratara de una matriz en parte de un programa, empleando la orden LET; a continuación damos un ejemplo:

```
30 LET A(1,2) = 3 800
40 LET A(1,3) = 3 500
50 LET A(1,4) = 52 700
.
.
.
610 LET A(12,5) = 8 600
```

Pero evidentemente ésta es una forma muy laboriosa de realizar el programa. Un procedimiento mucho más simple es emplear las órdenes READ y DATA, o bien INPUT junto con bucles FOR...NEXT intercalados.

Así, pues, veamos cómo podría hacerse utilizando la orden READ:

```
10 DIM A(12,5)
20 FOR F = 1 TO 12
30 FOR C = 1 TO 5
40 READ A(F,C)
50 NEXT C
60 NEXT F
70 DATA 20 000, 3 800, 3 500, 52 700, 8 300,
   20 000, 3 500, 2 900
80 DATA 51 000, 6 800, 20 000, 3 400, 3 100,
   49 900, 7 000, 20 000
90 DATA 3 300, 3 000, 50 000, 9 100, 20 000, 3 400,
   4 300, 47 000
100 DATA 7 900, 20 000, 3 600, 3 000, 48 600,
    6 100, 20 000, 3 500
110 DATA 3 200, 47 000, 8 500, 20 000, 3 400,
    3 100, 49 500, 7 400, 20 000
120 DATA 3 300, 3 000, 46 300, 6 900, 20 000,
    3 700, 3 800
130 DATA 52 800, 7 000, 20 000, 3 900, 3 800,
    53 000, 7 500
140 DATA 20 000, 4 100, 4 700, 55 900, 8 600
150 END
```

En este programa existen varios puntos importantes a tener en cuenta. El primero de ellos es que DIM está justo al principio del programa. Una orden DIM debe efectuarse una sola vez en un programa y, por lo tanto, lo normal es colocarla cerca del principio o antes de realizar ningún bucle. El segundo punto a considerar es que hay dos bucles FOR...NEXT, uno para situar la "fila" y otro para hacer lo mismo con la "columna". Estos dos bucles no están uno a continuación del otro, sino que uno está alojado en el interior del otro. Ténganse en cuenta los límites elegidos. FOR F = 1 TO 12 incrementará el valor de la fila de 1 a 12; FOR C = 1 TO 5 aumentará el valor de la columna de 1 a 5.

Exactamente en el centro del segundo bucle —que está alojado en el interior del primero— está situada la orden READ. La parte decisiva del programa es la siguiente:

```
20 FOR F = 1 TO 12
30 FOR C = 1 TO 5
40 READ A(F,C)
50 NEXT C
60 NEXT F
```

La primera vez que se procese, después de haber realizado las líneas 20 y 30, los valores asignados a F y C serán ambos 1, y, por lo tanto, la línea 40 será equivalente a READ A(1,1). La primera cantidad de la orden DATA es 20 000, y, en consecuencia, este valor será asignado a la primera fila y a la primera columna de la matriz.

Después que haya sucedido esto, NEXT C hace retroceder el programa a la línea 30 y el valor asignado a C pasa a ser 2. Ahora la línea 40 es equivalente a READ A(1,2) y la siguiente cantidad, 3 800, será asignada a la primera fila y segunda columna de la matriz. Este proceso se repite hasta que C haya alcanzado el valor 5. Después de ello, la orden NEXT F de la línea 60 hace retroceder el programa hasta la línea 20 y F toma el valor 2. La línea 30 ajusta otra vez el valor de C a 1 y, por consiguiente, ahora la línea 40 será equivalente a READ A(2,1).

De esta forma, los bucles anidados en esta manera son muy útiles, pero es necesario poner mucha atención. Cada bucle debe estar alojado completamente en el interior del otro y debe observarse cuidadosamente la correlación de las órdenes NEXT. Obsérvese cómo el primer bucle, FOR F, incluye la segunda orden NEXT. Cuando existen dos bucles, uno de ellos alojado en el interior del otro, el primero recibe el nombre de bucle exterior, y el segundo, el de bucle interior. Todo el bucle interior será siempre completado antes de que sea incrementado el índice del bucle exterior. Es posible introducir bucles en el interior de otros hasta la "profundidad" requerida por el programa, pero éste puede resultar complejo y difícil de seguir. Es una mala práctica colocar en el interior de los bucles instrucciones que hagan desviar el programa principal; asimismo debe evitarse emplear GOTO.

Veamos la orden DATA. Obsérvese que, para separar los datos de las cantidades, se emplean comas, pero que no tiene que haber coma antes de la primera cantidad o después de la última. Entre cada cantidad se han insertado espacios, pero esto no es lo normal. Al introducir los datos, es fácil cometer errores, que luego son difíciles de descubrir.

Se pueden utilizar tantas órdenes DATA como sean necesarias. Cada línea nueva debe empezar con una DATA. Cada vez se lee una cantidad, empezando por el principio de la primera DATA y continuando hasta que se hayan leído todas. Compruébese que el número de datos sea correcto, pues, en caso contrario, al pasar el programa se obtendrá una información errónea.

El programa presentado hasta ahora, realmente, no hace más que transformar los datos y disponerlos en forma de una matriz bidimensional. Después de introducir el programa y pulsar RUN, aparentemente no sucederá nada, y lo único que se verá en la pantalla será el esquema BASIC. Para comprobar que los datos están bien situados, deben probarse algunas órdenes PRINT. (En BASIC, una orden es una palabra clave que puede ejecutarse inmediatamente, sin tener que utilizar el programa, y, por tanto, no necesita un número de línea. Por ejemplo, LIST, RUN, SAVE, AUTO, EDIT y PRINT.) PRINT A(1,1) <CR> hará que el número 20 000 aparezca en la pantalla. ¿Qué se imprimirá con las siguientes órdenes?

```
PRINT A(12,1)
PRINT A(1,5)
PRINT A(5,1)
PRINT A(5,5)
```

Para lograr que el programa sea de utilidad, es necesario ampliarlo. En su forma actual constituye una base adecuada para un "programa principal". Para usarlo como parte de un programa mayor y más útil, pueden escribirse módulos como si constituyeran subrutinas, que serán empleadas por diversos GOSUB, insertados en puntos adecuados antes de la orden END.

En los primeros pasos del diseño de un programa de gastos domésticos, es mejor empezar con una simple descripción escrita de los requisitos generales. Puede decidirse que se quiere obtener el total de gastos y los promedios por meses, o bien para cada categoría (electricidad, por ejemplo). Se pueden desarrollar los detalles de cómo derivar estos resultados a una etapa posterior. Si se tiene que efectuar una elección en el programa sobre qué subrutinas se desea que sean realizadas, probablemente se requerirá la guía de un "menú", que controle directamente las subrutinas apropiadas. Un primer bosquejo del programa, en

esta fase de desarrollo, puede ser semejante al siguiente:

```
PROGRAMA PRINCIPAL
(ENTRADA DE DATOS)

MENU
(SELECCION DE SUBRUTINAS)

END
```

Al desarrollar el programa con algo más de precisión, se verá la necesidad de emplear subrutinas para calcular totales por meses (TOTALMES) o por categorías (TOTALCAT), los gastos promedios mensuales (PROMMES) y los promedios anuales por categorías (PROMCAT). La razón de utilizar nombres de una sola palabra para denominar estas subrutinas es ayudar a planear el programa sin tener que preocuparse, en esta fase, de detalles como dar un número de línea. Si lo pensamos detenidamente, quizá decidamos que incluso la parte principal del menú del programa debería tratarse como una subrutina con el fin de mantener la sección principal del programa como un módulo independiente. La siguiente etapa para perfilar el programa constará de los siguientes pasos:

```
PROGRAMA PRINCIPAL (ENTRADA DE DATOS)
MENU (LLAMADA A SUBRUTINAS)
END

**SUBRUTINAS**

1 MENU
2 TOTALES
3 PROMEDIOS

(2) TOTALES
4 TOTALMES
5 TOTALCAT

(3) PROMEDIOS
6 PROMMES
7 PROMCAT
```

Este esquema de programa muestra que la subrutina de MENU ofrece una elección entre TOTALES y PROMEDIOS. Ambas serán subrutinas. La subrutina de TOTALES ofrece una nueva elección entre TOTALMES y TOTALCAT. Éstas serán las subrutinas que efectuarán los cálculos reales.

La subrutina de PROMEDIOS permite elegir entre PROMMES y PROMCAT, las cuales, asimismo, constituyen las subrutinas que efectuarán los cálculos adecuados. En este punto, debe ser posible ver si el "programa" hará lo que se le pide, sin desarrollar ningún código real (programa detallado escrito en BASIC). Si hasta aquí se han realizado todas las etapas correctamente, ya se está en condiciones de abordar la escritura de los módulos (subrutinas). El único cambio que debe hacerse en el programa principal será una llamada de subrutina antes de END, es decir, habría que añadir:

```
145 GOSUB **MENU**
```

Obsérvese que se continúa empleando "nombres" para designar las subrutinas, en vez de asignarles números de línea. Numerosos lenguajes, el PASCAL, por ejemplo, permiten designar con nombres a los subprogramas, pero, en cambio, ello no es posible en la mayoría de las versiones de BASIC, que, por el contrario, necesitan utilizar números de línea. Sin embargo, tomaremos en consideración estos detalles más adelante.

Veamos cómo puede escribirse la subrutina de MENU (se han omitido los números de línea pero el usuario puede añadirlos si desea realizar este programa).

```
REM SUBRUTINA **MENU**
PRINT "¿DESEA T(OTALES) O P(ROMEDIOS)?"
PRINT "DIGITAR T O P"
INPUT L$
IF L$ = "T" THEN GOSUB *TOTALES*
IF L$ = "P" THEN GOSUB *PROMEDIOS*
RETURN
```

Nota: Las llamadas a subrutinas se están acotando entre asteriscos: *— —*. Al programar, en vez de ello, se deberán emplear números de línea, que podrán incorporarse cuando se esté en condiciones de conocer cuáles son.

Imaginemos que se teclea T, inicial de TOTALES. Entonces el programa pedirá la subrutina de TOTALES. Ésta presentará otro menú, semejante al siguiente:

```
REM SUBRUTINA DE **TOTALES**
PRINT "¿DESEA TOTALES DE?"
PRINT "¿M(ES) O C(ATEGORIA)?"
PRINT "DIGITAR M O C"
INPUT L$
IF L$ = "M" THEN GOSUB *TOTALMES*
IF L$ = "C" THEN GOSUB *TOTALCAT*
RETURN
```

Supongamos que se elige M, indicativo de TOTALMES. Veamos cómo se puede escribir un módulo para calcular los gastos totales para cualquiera de los meses del año.

```
REM SUBRUTINA DE **TOTALMES**
REM CALCULA LOS GASTOS TOTALES DE
REM CUALQUIER MES
PRINT "ELEGIR MES"
PRINT "1-ENE 2-FEB 3-MAR 4-ABR 5-MAYO"
PRINT "6-JUN 7-JUL 8-AGO 9-SEPT"
PRINT "10-OCT 11-NOV 12-DIC"
PRINT "DIGITAR EL NUMERO DEL MES"
LET T = 0
INPUT M
FOR C = 1 TO 5
LET T = T + A(M,C)
NEXT C
PRINT "LOS GASTOS TOTALES DEL MES"
PRINT "NUMERO";M;"SON";T
RETURN
```

Se digita el número que representa el mes, y la orden INPUT asigna el número a la variable M(MES). M se emplea para especificar el subíndice de la "fila" de la matriz bidimensional A. El bucle FOR-NEXT incrementa el valor de C (columna) desde uno hasta cinco y, por consiguiente, la primera vez que se pasa el bucle, si se había elegido tres, es decir, el mes de marzo, la orden LET será equivalente a LET T = T + A(3,1). La próxima vez que se utilice el bucle, será equivalente a LET T = T + A(3,2), y así sucesivamente.

En este capítulo le encomendamos a usted la tarea de escribir las otras subrutinas o tratar de solucionar los otros ejercicios. Las matrices bidimensionales son ideales para cualquier programa que contenga tablas

Respuestas a los "Ejercicios" de la página 175

Función RND

```
40 IF R > 6 THEN LET R = 1
```

Bucle y promedio

```
5 FOR L = 1 TO 100
:
:
80 LET T = T + R
90 NEXT L
100 LET A = T/100
110 END
```

Sustitución por una subrutina

Suprima las líneas 5, 80, 90, 100 y 110 en la solución anterior. Cambie las líneas 10 a 70 por (pongamos por caso) las 1000 a 1070. Verifique que la línea 40 es igual que en la función RND de la solución anterior. Luego añada 1080 RETURN. Incorpore el resultado en el programa principal. Cambie las líneas 50 y 130 para que se lea 50 GOSUB 1000 y 130 GOSUB 1000.

INKEY$

```
10 PRINT "PULSE CUALQUIER TECLA"
20 LET A$ = INKEY$
30 IF A$ = " " THEN GOTO 20
40 PRINT "LA TECLA QUE USTED HA PULSADO ES";A$
50 END
```

Bucle de tiempos

```
5 PRINT "PULSE LA BARRA ESPACIADORA DESPUES DE
  10 SEGUNDOS"
10 FOR L = 0 TO 1
20 LET R = R + 1
30 IF INKEY$ = " " THEN GOTO 60
40 LET L = 0
50 NEXT L
60 PRINT "EL VALOR DE R DESPUES DE
  10 SEGUNDOS ES";R
70 END
```

Comparaciones IF... THEN

```
10 GOSUB 1000
20 PRINT "ADIVINE EL NUMERO"
30 FOR G = 1 TO 50
40 INPUT N
50 IF N > R THEN GOTO 110
60 IF N < R THEN GOTO 130
70 IF N = R THEN GOTO 150
80 NEXT G
90 PRINT "YA NO TIENE MAS OPORTUNIDADES.
   ¡HA PERDIDO!"
100 GOTO 500
110 PRINT "SU ELECCION ES DEMASIADO
    ELEVADA"
120 GOTO 80
130 PRINT "SU ELECCION ES DEMASIADO BAJA"
140 GOTO 80
150 PRINT "HA ACERTADO, FELICITACIONES"
500 END
1000 REM **SUBRUTINA ALEATORIA**
     (Introduzca aquí su subrutina)
1020 RETURN
```

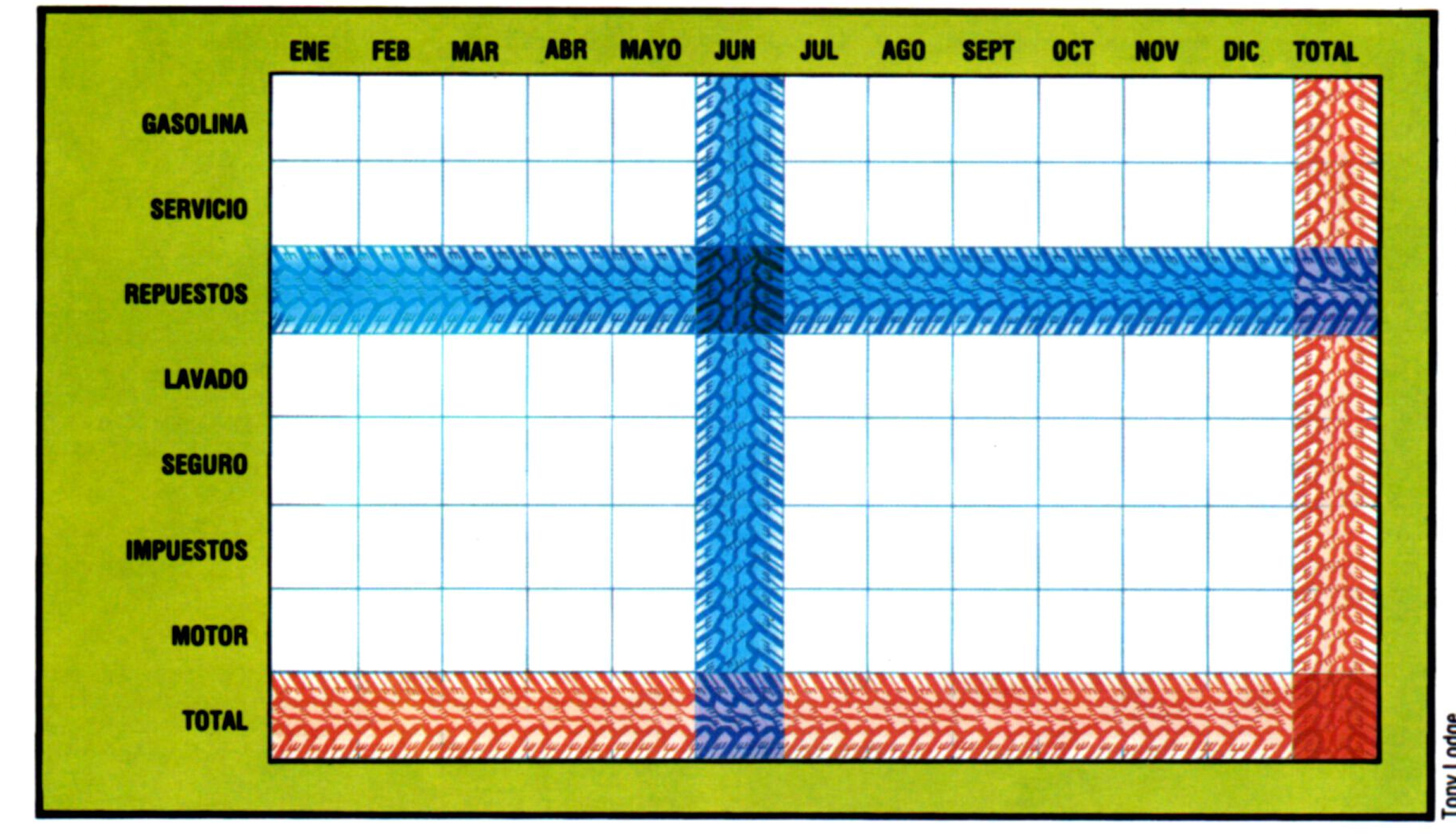
de datos, ya sean éstos estadísticos, financieros o de cualquier otro tipo.

Ejercicios

■ **Asignación de valores** Escriba un programa que asigne valores a los elementos ("gasolina", "servicio", etc.) de la matriz representada en la parte inferior de esta página. A continuación, escriba una subrutina que requiera un mes y uno de los elementos e imprima el contenido del cuadrado determinado por ambos. Por último, escriba una subrutina que calcule el total de cada columna y sitúe el resultado al pie del cuadrado, haga lo mismo con cada fila y calcule el total final, que deberá colocar en el cuadrado inferior derecho.

■ **Bugs** El programa siguiente no funcionaría adecuadamente y produciría un mensaje erróneo. Descubrir el error y efectuar la corrección pertinente.

```
10 DIM A(3,4)
20 FOR F = 1 TO 3
30 FOR C = 1 TO 4
40 READ A(F,C)
50 NEXT C
60 NEXT F
70 FOR X = 1 TO 3
90 FOR Y = 1 TO 4
100 PRINT A(Y,X)
110 NEXT Y
120 NEXT X
130 DATA 2,4,6,8,10,12,14,16,18,20,22
140 END
```



Tony Lodge

Gastos de un coche
La figura muestra una cuadrícula de 8 × 13 cuadrados. Las filas representan elementos diferentes del coste de mantenimiento de un coche, y las columnas los diferentes meses del año. Realizar el ejercicio "Asignación de valores" para calcular el coste anual de mantenimiento de un coche

Trazadores de gráficos

Mediante estos dispositivos, un ordenador puede producir gráficos de alta calidad. Funcionan con plumas de punta de fibra y algunos de ellos pueden cambiar automáticamente de color

La capacidad de crear copias impresas de los diagramas que aparecen en la pantalla de un ordenador es un requisito esencial para muchas personas que utilizan la máquina profesionalmente. Ingenieros, científicos, diseñadores y hombres de negocios necesitan diagramas y cuadros con un grado de precisión que no pueden suministrar las impresoras convencionales. El único dispositivo que puede crear esas imágenes es un trazador de imágenes, y, hasta hace poco tiempo, éste resultaba excesivamente caro para el usuario del ordenador personal.

Sin embargo, con la introducción de dispositivos como el trazador de gráficos-impresora de cuatro plumas utilizado por el Tandy/CGP-115 y el Oric MCP-40, un terminal de gráficos puede estar al alcance de una gran parte de los usuarios. Recientemente ha aparecido en el mercado una gama completa de trazadores de gráficos con unas características que antes sólo podían ofrecer al usuario máquinas cuyo precio era prohibitivo.

La necesidad de utilizar un trazador de gráficos, o *plotter*, generalmente está determinada por el tipo de trabajo a que está destinado el ordenador. Un ingeniero o un proyectista necesitarán dibujos precisos de equipos y montajes, en cambio un hombre de negocios deseará cuadros y gráficos que muestren los volúmenes de ventas. Realizar esto con impresoras convencionales es un proceso muy laborioso y los resultados sólo aparecerán en blanco y negro. La otra única opción de bajo coste es efectuar una fotografía de la pantalla, pero si bien esto puede ser suficiente para gráficos comerciales, no reúne los requisitos mínimos de precisión que pedirían un arquitecto o un diseñador.

Los trazadores de gráficos funcionan de una forma completamente distinta a las impresoras: trazan líneas entre dos puntos en lugar de partir de formas preestablecidas o modelos de puntos.

El principio básico con el que funcionan todas las marcas consiste en un sistema de coordenadas X, Y. Al igual que una gráfica puede ser trazada definiendo las coordenadas por las que debe pasar la línea, también una figura puede ser descompuesta en una serie de coordenadas. Para poder unir estas coordenadas con el fin de recrear la figura, tiene que existir alguna forma de movimiento. Por ello se fija la pluma a un pantógrafo que puede desplazarse en el sentido de las abscisas X (de izquierda a derecha) al mismo tiempo que la pluma se mueve a lo largo del pantógrafo en el sentido de las ordenadas Y (de arriba abajo).

El tipo tradicional de trazador de gráficos se conoce con el nombre de *lecho plano*, debido a que el papel es fijado a una placa plana, sobre la cual se desplaza el pantógrafo, tal como se puede apreciar en la ilustración. Esto tiene el inconveniente de que el trazador debe ser, como mínimo, tan grande como la hoja de papel.

Soporte pluma
Se pueden cambiar automáticamente hasta tres plumas. El pantógrafo regresa al soporte y cambia la pluma en uso por el nuevo color necesario. Manualmente se pueden cambiar más colores

Clips magnéticos
Mantienen el papel plano sobre el trazador. Están hechos de un material magnético flexible

Portaplumas
La pluma que se está utilizando se fija (en este caso magnéticamente) en el soporte, que se desplaza hacia abajo y sitúa la pluma en contacto con el papel

Un método para reducir su tamaño es adoptar una versión a gran escala del modelo del trazador de cuatro plumas (véase ilustración), en el cual el papel se desplaza en una dirección y las plumas en la contraria. A este tipo pertenecen los de las marcas Strobe 100 y Hewlett Packard Sweetlips. El movimiento del papel debe ser controlado con la misma precisión que el desplazamiento del pantógrafo en los del tipo de lecho plano, y esto se logra con un motor paso a paso. Este tipo de motor posee unas características muy especiales, y cada vez que recibe un pulso de la potencia aplicada, realiza una fracción de giro. Se aplica principalmente a unidades de disco, en las cuales controla la posición del cabezal sobre la superficie del disco, y en robótica (véase p. 176).

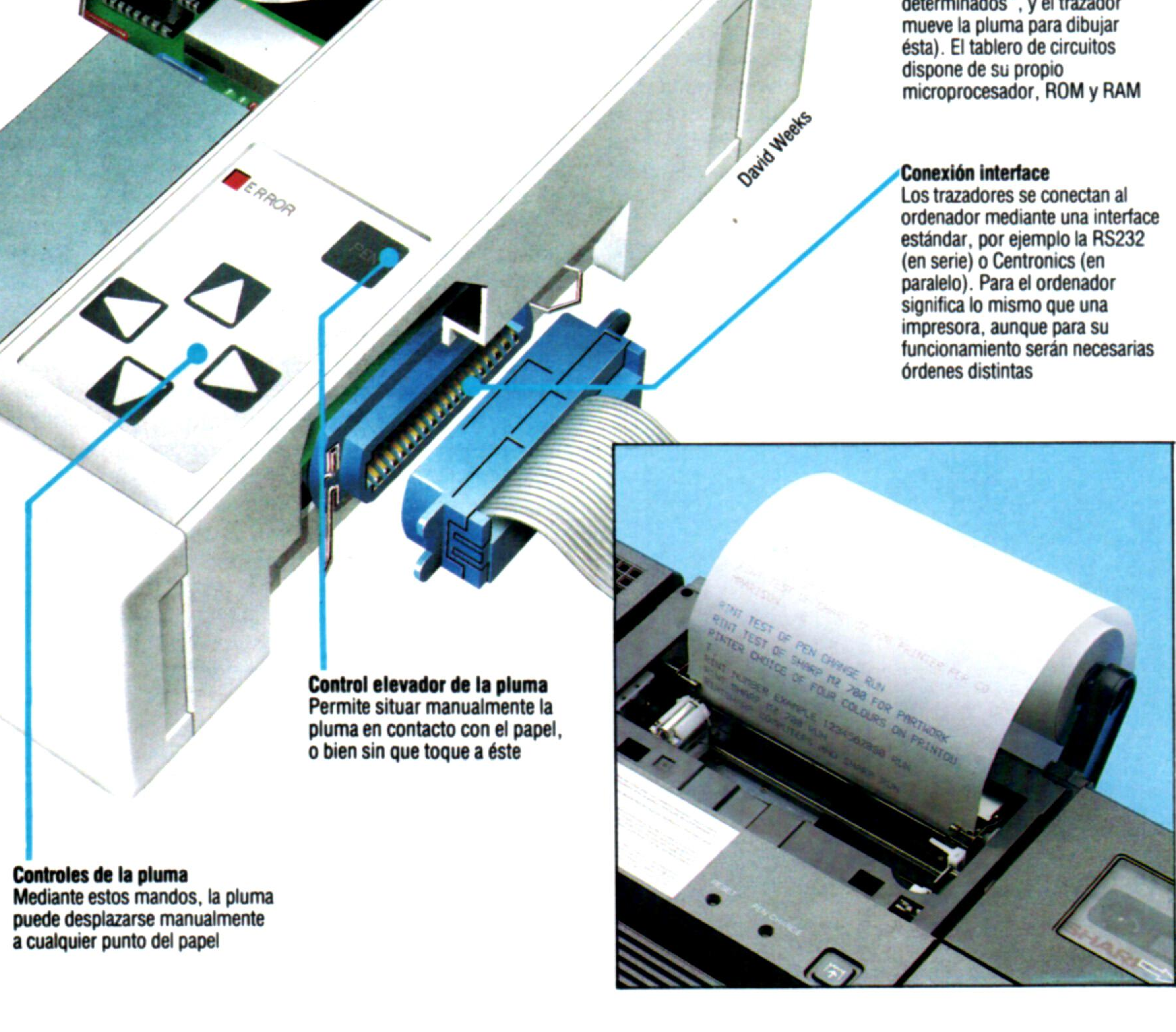
La conexión de un trazador a un ordenador es, por lo general, similar a conectar una impresora, por lo menos en lo que se refiere a la interface. Normalmente, pueden adquirirse trazadores con interfaces en serie (RS232) o en paralelo (Centronics o IEEE488),

que pueden enchufarse a la conexión utilizada por las impresoras. A menudo, su programación es algo más complicada. En vez de sólo enviarle los resultados de un programa que debe ser impreso, ahora será necesario suministrarle también información sobre cómo deben ser presentados. Por lo común, esto se realiza de una forma muy parecida a la empleada para formar un diagrama en la pantalla.

Debido al complicado sistema que utilizan los trazadores para realizar su cometido, éstos son, por regla general, "inteligentes". Esto quiere decir que poseen microprocesadores que convierten los caracteres e instrucciones recibidos del ordenador en una serie de coordenadas, que luego dibuja el trazador. Muchos de los más perfeccionados permiten también dibujar figuras complejas, tales como círculos y curvas, proporcionándoles únicamente los puntos de partida: el trazador hará el resto. El rotulado de los gráficos y diagramas y el coloreado de diagramas de barras y de segmentos circulares son, frecuentemente, procesos automáticos, factor que hace la programación mucho más sencilla.

Muchos trazadores se suministran junto con el software, lo cual les permite estar conectados directamente al interior del programa, en vez de ser como una copia sobre papel de la pantalla. Si no se dispone de este tipo de programas, el usuario debe desarrollar las rutinas necesarias para transformar la información de la pantalla en los códigos adecuados para guiar al trazador. Algunos de ellos no disponen de juegos de caracteres y, por consiguiente, deben crearse incluso los códigos para las letras y números. En cambio, esto permite al usuario crear sus propios caracteres y tipos.

Una vez que se ha generado una figura, ésta puede trazarse en cualquier posición, orientación o tamaño, con lo cual puede obtenerse una colección de formas para diferentes usos. Con frecuencia son muy útiles las rutinas para trazar círculos, curvas y figuras en secciones de gráficas, especialmente en el campo de los gráficos comerciales. Sin embargo, los principios de creación de un dibujo a partir de las coordenadas de la pantalla son exactamente los mismos que para crear la figura sobre el papel y, por lo tanto, la programación es bastante sencilla.



Pantógrafo
Puede situarse en cualquier punto a lo largo de la hoja (eje de las X) y el portaplumas es desplazado según el eje de las Y. La combinación de los movimientos de izquierda a derecha y de atrás adelante permite llegar a cualquier punto de la página

Motores paso a paso
Estos motores giran sólo unos grados por cada pulso eléctrico. Accionados adecuadamente, proporcionan el movimiento preciso de la pluma y el pantógrafo

Tablero de circuitos
Normalmente, los plotters son dispositivos "inteligentes" (pueden dárseles órdenes de alto nivel, por ejemplo: "dibujar una circunferencia de radio y centro determinados", y el trazador mueve la pluma para dibujar ésta). El tablero de circuitos dispone de su propio microprocesador, ROM y RAM

Conexión interface
Los trazadores se conectan al ordenador mediante una interface estándar, por ejemplo la RS232 (en serie) o Centronics (en paralelo). Para el ordenador significa lo mismo que una impresora, aunque para su funcionamiento serán necesarias órdenes distintas

Control elevador de la pluma
Permite situar manualmente la pluma en contacto con el papel, o bien sin que toque a éste

Controles de la pluma
Mediante estos mandos, la pluma puede desplazarse manualmente a cualquier punto del papel

Trazador de gráficos-impresora de cuatro plumas

Este mecanismo atrajo la atención de la industria cuando apareció por primera vez en la impresora Sharp CE-150. Sus hermanos mayores: el Tandy's CGP-115 y el Oric MCP-40, ayudaron a que el coste de las impresoras en color fuera asequible para el usuario del ordenador doméstico.
Como todas las buenas ideas, el sistema es conceptualmente muy simple. Un rollo de papel es empujado a través del mecanismo por medio de un piñón. El papel es desplazado, hacia adelante y hacia atrás, en intervalos muy precisos, mientras un portaplumas que sujeta los cursores de los cuatro colores se desplaza por la superficie, de izquierda a derecha y viceversa.
El portaplumas, para crear el texto o el gráfico, gira hasta enfrentar el color correcto y luego la pluma es presionada contra el papel. Las líneas horizontales se obtienen por el desplazamiento de la pluma y con el papel inmóvil; las verticales, con el papel en movimiento y la pluma estática. Las combinaciones de ambos movimientos producen diagonales o curvas. La calidad de la impresión es notable, pero la limitada anchura del papel la hace inadecuada para procesamiento de textos u otros fines profesionales

Alan Turing

El test para determinar la inteligencia de una máquina tomó su nombre de este matemático británico. La mayor parte de su trabajo, sin embargo, lo realizó para el servicio de inteligencia militar durante la última guerra mundial

Ciencia y deporte
Alan Turing (1912-1954) se inspiraba y relajaba mediante las carreras de fondo. Estaba intrigado por el efecto del esfuerzo físico sobre la creatividad y agilidad mental

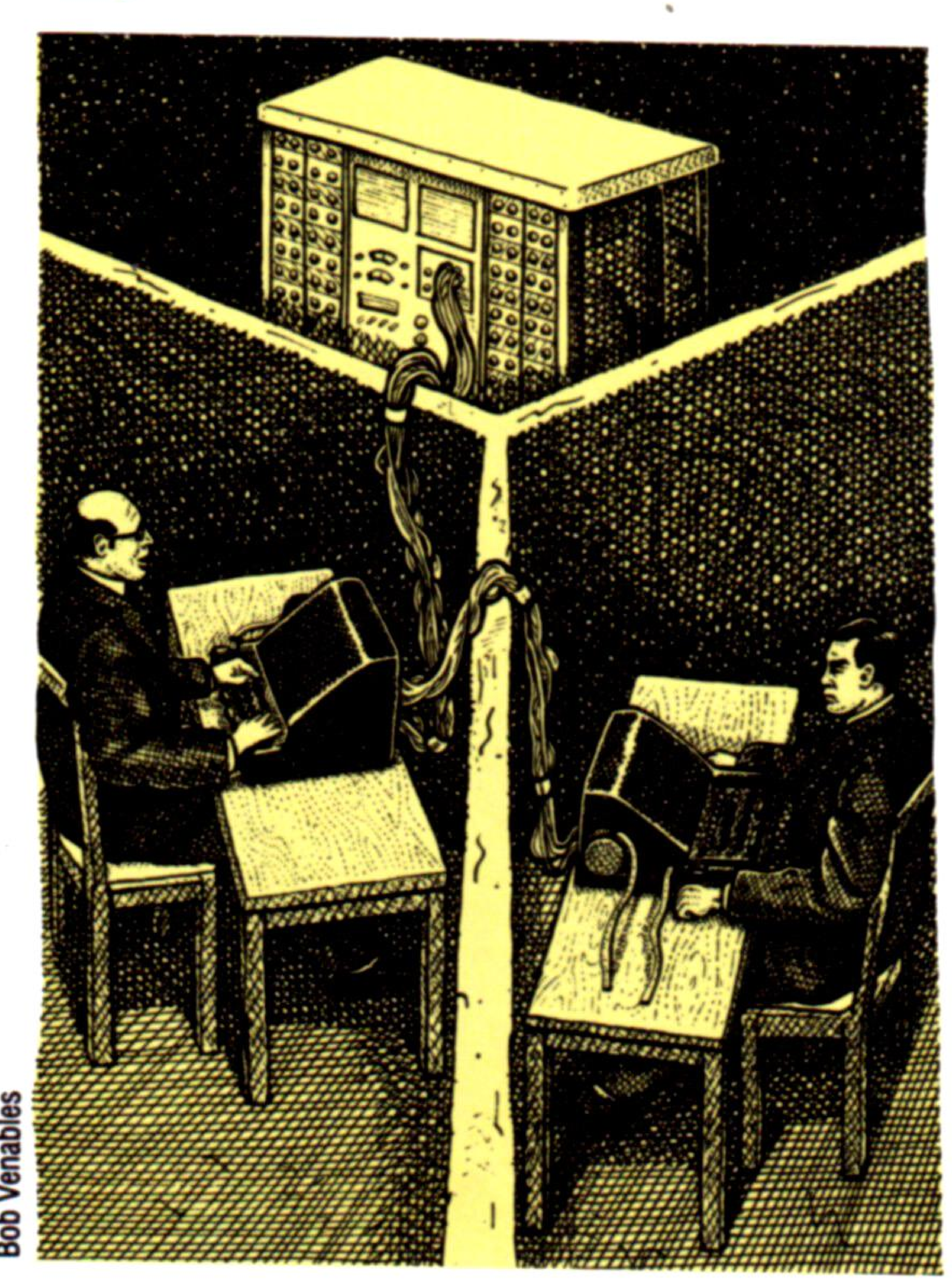


El joven Alan Turing mostró una extraordinaria intuición científica. Estando en la escuela, escribió a su madre: "Me parece que siempre quiero hacer las cosas a partir de aquello que abunda más en la naturaleza". Los matemáticos ofrecen pronto muestras de su talento, y Turing, tan pronto supo leer y escribir, empezó a componer versos y a diseñar bicicletas anfibias.

Mientras su padre se hallaba en Madrás, trabajando para el Indian Civil Service, Turing ganó numerosos premios escolares y más tarde una beca que le llevaría al King's College de Cambridge. Durante su permanencia en esta ciudad, primero como estudiante y luego como miembro de la junta de gobierno de su colegio, fue cuando su interés empezó a centrarse en los problemas de la lógica matemática.

En 1931, el matemático checo Kurt Gödel asombró al mundo científico con el descubrimiento de que había teoremas matemáticos que eran verdaderos aunque nunca pudieran probarse. Alan Turing se puso a investigar aquellos que podían ser probados.

Propuso una máquina, cuya construcción dejó a la imaginación, que podría realizar de manera mecánica los procesos que por regla general lleva a cabo un matemático. Para cada proceso había una máquina: por ejemplo, una máquina para sumar, otra para dividir, y una tercera para integrar, y así sucesivamente. Más tarde estas máquinas serían conocidas como máquinas Turing.



Bob Venables

¿Pueden pensar las máquinas?
Para contestar a esta pregunta, Turing propuso su famoso test llamado *juego de la imaginación*, y conocido también como *test de Turing*. Una persona se sitúa en una habitación con una teleimpresora, la cual está conectada a otra teleimpresora, colocada en una habitación diferente y manejada también por un operador, así como al ordenador que se quiere probar. La primera persona puede hacer cualquier pregunta que desee a uno de los dos. Si, consecuentemente, ésta es incapaz de determinar cuándo está en comunicación con la otra persona y cuándo con el ordenador, la máquina puede ser considerada como inteligente. Después de todo, continúa la tesis, no tenemos forma de asegurar si otras personas razonan o son conscientes, excepto comparando sus reacciones con las nuestras propias frente a un hecho determinado

El joven matemático investigó los trabajos de estas máquinas imaginarias y llegó a una notable conclusión. En vez de destinar la realización de cada proceso matemático a una máquina independiente, era posible diseñar un aparato "universal" que fuera capaz, al ser "programado", de cumplir las funciones de cualquiera de las máquinas especializadas. Alan Turing había encontrado así, casualmente, la teoría del ordenador programable.

Al estallar la segunda guerra mundial, Turing fue rápidamente alejado del mundo académico y reclutado por la Escuela de Códigos y Cifrados del gobierno británico, situada en Bletchley Park (Buckinghamshire). Si no hubiera sido por el conflicto bélico, quizá sus máquinas hubieran continuado siendo una entelequia.

Las actividades que se realizaban en Bletchley Park, que eran altamente secretas, consistían de manera primordial en descifrar con suma urgencia los códigos militares alemanes.

Debido a que estos códigos podían ser cambiados diariamente, era necesario el empleo de máquinas para su desciframiento antes de que los servicios alemanes introdujeran nuevos códigos. Bletchley Park se convirtió en un inmenso centro de procesamiento de información. Promediada la guerra, Turing fue enviado a Estados Unidos, con el fin de que creara unos códigos seguros para las comunicaciones trasatlánticas entre los países aliados.

Debido al carácter secreto de su trabajo en esta época, se dispone de muy pocos datos sobre su actividad. Sin embargo, se cree fundamentalmente que conoció a Von Neumann cuando se hallaba en Princeton (Nueva Jersey). Hacia el final de la guerra, Turing recibió el encargo de empezar los trabajos para la construcción de un ordenador totalmente británico, destinado al National Physical Laboratory, y que recibiría el nombre de ACE (Automatic Computing Engine).

Esta máquina tardó mucho tiempo en ser construida, pero muchas de sus características eran superiores a las del ENIAC (véase p. 46). Frustrado por el lento avance, Turing dimitió y se fue a vivir a Manchester, donde empezó a colaborar en el proyecto del ordenador de la universidad. Al mismo tiempo, era asesor de la compañía Ferranti y, por tanto, colaboró en la construcción de los primeros ordenadores fabricados en Gran Bretaña.

Turing fue una personalidad excéntrica, que se dedicó a lo que sabía que era importante sin preocuparse de prejuicios sociales o cortapisas legales. Un amigo dijo de él que era "divinamente retrasado" para ver los errores de los otros, pero su genio científico era intachable. En 1952 fue condenado por acusaciones relativas a homosexualidad y, dos años más tarde, se suicidó. ¿Quién puede predecir cuál hubiera sido su contribución al desarrollo de la inteligencia artificial si no hubiera adoptado esa trágica determinación?

DONDE CONSEGUIR TU sinclair

ALAVA
COMPONENTES ELECTRONICOS GAZTEIZ
Domingo Beltran, 58 (Vitoria)
DEL CAZ
Avda. Gazteiz, 58 (Vitoria)
VALBUENA
Virgen Blanca, 1 (Vitoria)

ALBACETE
ELECTRO MIGUEL
Tesifonte Gallego, 27
TECON
Maria Marin, 13

ALICANTE
ASEMCA (Villena)
Avda. de la Constitucion, 54 (Villena)
CONSULTING DESARROLLO INFORMATICO
Pais Valencia, 54 (Alcoy)
COMPONENTES ELECTRONICOS LASER
Jaime M.ª Buch, 7
ELECTRODATA LEVANTE
San Vicente, 28
ELECTRONICA AITANA
Limones, s/n. Edificio Urgull (Benidorm)
ELECTRONICA OHMIO
Avda. El Hamed, 1
LIBRERIA LLORENS
Alameda, 50 (Alcoy)

AVILA
FELIX ALONSO
San Segundo, 15

BADAJOZ
MECANIZACION EXTREMEÑA
Vicente Barantes, 18
SONYTEL
Villanueva, 16

BARCELONA
ARTO
C/ Angli, 43
BERENGUERAS
C/ Diputacion, 219
CATALANA D'ORDINADORS
C/ Trafalgar, 70
CECSA
C/ Mallorca, 367
COMPUTERLAND
C/ Infanta Carlota, 89
COMPUTERLAND
Trav. de Dalt, 4
COPIADUX
C/ Dos de Mayo, 234
D. P. 2000
C/ Sabino de Arana, 22-24
DIOTRONIC
C/ Conde Borrell, 108
EL CORTE INGLES
Avda. Diagonal, 617-619
EL CORTE INGLES
Pza. Cataluña, 14
ELECTRONICA H. S.
C/ S. José Oriol, 9
ELECTRONICA SAUQUET
C/ Guilleries, 10
ELEKTROCOMPUTER
Via Augusta, 120
EXPOCOM
C/ Villarroel, 68
GUIBERNAU
C/ Sepulveda, 104
INSTA-DATA
P.º S. Juan, 115
MAGIAL
C/ Sicilia, 253
MANUEL SANCHEZ
Pza. Major, 40 (Vic)
MILLIWATTS
C/ Melendez, 55 (Mataró)
ONDA RADIO
Gran Via, 581
RADIO ARGANY
C/ Borrell, 45
RADIO SONDA
Avda. Abad Margat, 77 (Tarrasa)
RAMEL ELECTRONICA
Cr. de Vic, 3 (Manresa)
REDISA GESTION
Avda. Sarriá, 52-54
RIFE ELECTRONICA
C/ Aribau, 80, 5.º, 1.ª
SERVICIOS ELECTRONICOS VALLES
Pza. del Gas, 7 (Sabadell)
SISTEMA
C/ Balmes, 434
S. E. SOLE
C/ Muntaner, 10
SUMINISTROS VALLPARADIS
C/ Dr. Ferrén, 172 (Tarrasa)
TECNOHIFI, S. A.
C/ La Rambleta, 19
VIDEOCOMPUT
P.º Pep Ventura, 9, Bl. C, Bjos. Bis (Vic)

BURGOS
COMELECTRIC
Calzada, 7
ELECTROSON
Conde don Sancho, 6

CACERES
ECO CACERES
Diego Maria Crehuet, 10-12

CADIZ
ALMACENES MARISOL
Camoens, 11 (Ceuta)
INFORSA
Avda. Fuerzas Armadas, 1 (Algeciras)
ELECTRONICA VALMAR
Ciudad de Santander, 8
M. R. CONSULTORES
Multi Centro Merca 80 (Jerez de la Frontera)
PEDRO VAREA
Porvera, 36 (Jerez de la Frontera)
LEO COMPUTER
Garcia Escamez, 3
SONYTEL
Queipo de LLano, 17
SONYTEL
Jose Luis Diez, 7
T L C Y AUTOMATICA
Dr. Herrera Quevedo, 2

CASTELLON
NOÙ DESPACH'S
Rey D. Jaime, 74

CIUDAD REAL
COMERCIAL R. P.
Travesera de Coso, 2 (Valdepeñas)
ECO CIUDAD REAL
Calatrava, 8

CORDOBA
ANDALUZA DE ELECTRONICA
Felipe II, 15
CONTROL
Conde de Torres Cabrera, 9
ELECTRONICA PADILLA
Sevilla, 9
MORM
Plaza Colón, 13
SONYTEL
Arte, 3
Avda. de los Mozárabes, 7

CUENCA
SONYTEL
Dalmacio Garcia Izcara, 4

GERONA
AUDIFILM
C/ Albareda, 15
CENTRE DE CALCUL DE CATALUNYA
C/ Barcelona, 35
S. E. SOLE
C/ Sta. Eugenia, 59

GRANADA
INFORMATICA Y ELECTRONICA
Melchor Almagro, 8
SONYTEL
Manuel de Falla, 3
TECNIGAR
Ancha de Gracia, 11

GRANOLLERS
COMERCIAL CLAPERA
C/ Maria Maspons, 4

GUIPUZCOA
ANGEL IGLESIAS
Sancho el Sabio, 7-9
BHP NORTE
Ramón M.ª Lili, 9
ELECTROBON
Reina Regente, 4

HUELVA
SONYTEL
Ruiz de Alda, 3

HUESCA
ELECTRONICA BARREU
M.ª Auxiliadora, 1

IBIZA
IBITEC
C/ Aragón, 76

JAEN
CARMELO MILLA
Coca de la Piñera, 3
MARA ILUMINACION
Avda. Linares, 13 (Ubeda)
MICROJISA
Garcia Rebull, 8
SONYTEL
José Luis Diez, 7
SONYTEL
Pasaje del Generalisimo, 3 (Linares)

LA CORUÑA
DAVIÑA
Republica de El Salvador, 29 (Santiago)
PHOTOCOPY
Teresa Herrera, 9
SONYTEL
Avda. de Arteijo, 4
SONYTEL
Tierra, 37

LAS PALMAS DE GRAN CANARIA
COMPUTERLAND
Carvajal, 4
CHANRAI
Triana, 3
EL CORTE INGLES
José Mesa y Lopez, 18

LEON
ELECTROSON
Avda. de la Facultad, 15
MICRO BIERZO
Carlos I, 2 (Ponferrada)
RADIO RACE
Modesto Lafuente, 3

LERIDA
SELEC
C/ Ferrer y Busquet, 14 (Mollerusa)
SEMIC
C/ Pi y Margall, 47

LUGO
ELECTROSON
Concepcion Arenal, 38
SONYTEL
Primo de Rivera, 30

MADRID
ALFAMICRO
Augusto Figueroa, 16
BELLTON'S
Torpedero Tucuman, 8
CHIPS-TIPS
Pto. Rico, 21
CMP
Pto. Santa Maria, 128
COMPUTERLAND
Castello, 89
COSESA
Barquillo, 25
DINSA
Gaztambide, 4
DISTRIBUIDORA MADRILEÑA
Todos sus centros
ELECTROSON
Duque de Sexto, 15 (y otros centros)
INVESMICROSTORE
Genova, 7
J.P. MICROCOMPUT
Montesa, 44
EL CORTE INGLES
Todos sus centros
ELECTRONICA SANDOVAL
Sandoval, 4
PENTA
Dr. Cortezo, 12
RADIO CINEMA
Antonio Acuña, 3
RADIO QUER
Todos sus centros
SONYTEL
Clara del Rey, 24 (y todos sus centros)
SONICAR
Vallehermoso, 19
VIDEOMUSICA
Orense, 28

MALAGA
EL CORTE INGLES
Prolongación Alameda, s/n.
INGESCON
Edificio Galaxia
SONYTEL
Salitre, 13

MELILLA
OFI-TRONIC
Hermanos Cayuela, 11

MENORCA
ELECTRONICA MENORCA
C/ Miguel de Veri, 50 (Mahon)

MURCIA
COMPUTER LIFE
Alameda San Antón, 2 (Cartagena)
EL CORTE INGLES
Libertad, 1
ELECTRONICA COMERCIAL CRUZ
Rio Segura, 2
MICROIN
Gran Via, 8

NAVARRA
ENER
Paulino Caballero, 39
GABINETE TECNICO EMPRESARIAL
Juan de Labrit, 3
JOSE LUIS DE MIGUEL
Arrieta, 11 bis

OVIEDO
AUTECA
Valentin Masip, 25
EDIMAR
Cangas de Onis, 4-6 (Gijón)
ELECTRONICA RATO
Versalles, 45 (Avilés)
RADIO NORTE
Uria, 20
RESAM ELECTRONICA
San Agustin, 12 (Gijón)
RETELCO
Cabrales, 31 (Gijón)
SELECTRONIC
Fermin Canellas, 3

ORENSE
SONYTEL
Concejo, 11

PONTEVEDRA
EL CORTE INGLES
Gran Via, 25 (Vigo)
ELECTROSON
Santa Clara, 32
ELECTROSON
Venezuela, 32 (Vigo)
SONYTEL
Salvador Moreno, 27
SONYTEL
Gran Via, 52 (Vigo)
TEFASA COMERCIAL
San Salvador, 4 (Vigo)

PALMA DE MALLORCA
GILFT
Via Alemania, s/n
IAM
C/ Cecilio Metlo, 5
TRON INFORMATICA
C/ Juan Alcover, 54, 6.º C

LA RIOJA
YUS COMESSA
Cigueña, 15

SALAMANCA
DEL AMO
Arco, 5
PRODISTELE
España, 65

SANTANDER
LAINZ S. A.
Reina Victoria, 127
RADIO MARTINEZ
Dr. Jimenez Diaz, 13

SEGOVIA
ELECTRONICA TORIBIO
Obispo Quesada, 8

SEVILLA
A.D.P.
San Vicente, 3
EL CORTE INGLES
Duque de la Victoria, 10
SCI
Aceituno, 8
SONYTEL
Pages del Corro, 173
Adriano, 32

TARRAGONA
AIA
Rambla Nova, 45, 1.º
CIAL. INFORMATICA TARRAGONA
C/ Gasometro, 20
ELECTRONICA REUS
Avda. Prat de la Riba, 5 (Reus)
SEIA
Rambla Vella, 7 B
S.E. SOLE
C/ Cronista Sese, 3
T. V. HUGUET
Pza. Major, 14 (Montblac)
VIRGILI
C/ Dr. Gimbernat, 19 (Reus)

STA. CRUZ DE TENERIFE
COMPUTERLAND
Méndez Nuñez, 104 B
TRENT CANARIAS
Serrano, 41

VALENCIA
ADISA
San Vicente, 33 (Gandia)
CESPEDES
San Jacinto, 6
COMPUTERLAND
Marqués del Turia, 53
DIRAC
Blasco Ibañez, 116
EL CORTE INGLES
Pintor Sorolla, 26
Melendez Pidal, 15
PROMOCION INFORMATICA
Pintor Zariñena, 12

VALLADOLID
SONYTEL
León, 4

VIZCAYA
BILBOMICRO
Aureliano del Valle, 7
DATA SISTEMAS
Henao, 58
DISTRIBUIDORA COM
Gran Via, 19-21 y todos sus centros
EL CORTE INGLES
Gran Via, 9
ELECTROSON
Alameda de Urquijo, 71
San Vicente, 18 (Baracaldo)
GESCO INFORMATICA
Alameda de Recalde, 76
KEYTRON
Hurtado de Amezaga, 20

ZAMORA
MEZZASA
Victor Gallego, 17

ZARAGOZA
EL CORTE INGLES
Sagasta, 3
SONYTEL
Via Pignatelli, 29-31



DISTRIBUIDOR EXCLUSIVO:
INVESTRONICA

Central Comercial
TOMAS BRETON, 60
TELF. 468 03 00
TELEX 23399 IYCO E
MADRID

Delegación Cataluña
MUNTANER, 565
TELF. 212 68 00
BARCELONA



10009
9 788485 822836